EURASIP Journal on Advances in Signal Processing

Digital Audio Effects

Guest Editors: Augusto Sarti, Udo Zoelzer, Xavier Serra,
Mark Sandler, and Simon Godsill

Digital Audio Effects

EURASIP Journal on
Advances in Signal Processing

Digital Audio Effects

Guest Editors: Augusto Sarti, Udo Zoelzer, Xavier Serra, Mark Sandler, and Simon Godsill

Contents

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 459654, 2 pages
doi:10.1155/2010/459654

Editorial
Digital Audio Effects

Augusto Sarti (EURASIP Member),[1] Udo Zoelzer,[2] Xavier Serra,[3] Mark Sandler,[4] and Simon Godsill (EURASIP Member)[5]

[1] *Dipartimento di Elettronica e Informazione (DEI), Politecnico di Milano, Milano, Italy*
[2] *Department of Signal Processing and Communications, Helmut-Schmidt-University—University of Federal Armed Forces Hamburg, Germany*
[3] *Music Technology Group, Department of Information and Communication Technologies & Audiovisual Institute, Universitat Pompeu Fabra, Barcelona, Spain*
[4] *Centre for Digital Music (C4DM), School of Electronic Engineering and Computer Science, Queen Mary University of London, UK*
[5] *Department of Engineering, University of Cambridge, UK*

Correspondence should be addressed to Augusto Sarti, sarti@elet.polimi.it

Received 31 December 2010; Accepted 31 December 2010

Digital audio effects usually refer to all those algorithms that are used for improving or enhancing sounds in any step of a processing chain of music production, from generation to rendering. Today these algorithms are widely used in professional or home music production studios, electronic or virtual musical instruments, and all kinds of consumer devices, including videogame consoles, portable audio players, smartphones, or appliances. Motivated by this expansion trend, in the past few years the range of research topics that have fallen within the digital audio effects realm has broadened to accommodate new topics and applications, from space-time processing to human-machine interaction.

All the technologies and the research topics that are behind such topics are today addressed by the International Digital Audio Effects Conference (DAFx), which has become a reference gathering for researchers working in the audio field. In the many editions of the DAFx conference, we have witnessed a proliferation of new and emerging methodologies for digital audio effects at many levels of abstraction, from signal level to symbol level. Some of the contributions to this special issue, in fact, are linked to works presented in this conference and seem to capture this transformational trend.

Two contributions of this special issue deal with aspects of sound synthesis. Synthetic sound generation is an important aspect of sound effects, whose importance has recently grown beyond the boundaries of musical sound synthesis. While virtual environments are becoming more and more part of our everyday life, the sonification of acoustic events in such environments is, in fact, still an open problem. The first contribution of the series, by C. Picard et al., addresses exactly this issue and provides analysis tools for determining the parameters of modal sound synthesis. The second contribution, by J. Pakarinen, offers a different set of analysis tools for parameter estimation, this time devoted to a hot topic in the DAFx community, which is that of virtual analog processing, with particular reference to the nonlinearities that characterize the reference analog systems that are being emulated. The third contribution, by A. Novak et al., allows us to take a different look at nonlinearities, this time with reference to audio effects for music production.

Digital audio effects are also part of the music production processing chain, which includes preprocessing, editing and mixing. The paper, by Terrell et al., is concerned with the noise gate, a specific type of digital effect, important for the capturing drum performances and dealing with bleeds from secondary sources. Another classical type of effects widely used in music production is time/pitch scaling. This effect is addressed in the contribution authored by E. Azarov et al. The paper by E. Perez et al. again addresses music production aspects, as it proposes a solution for automatic panning effects in music mixing.

Sound rendering and, particularly, spatial rendering are progressively gaining more and more importance in the research community of DAFx. In this line of work is the paper authored by F. Antonacci et al. which introduces a seminal

work on geometric wavefield decomposition which accounts for propagation phenomena such as diffusion and diffraction and serves as a computational engine for both wavefield rendering and binaural rendering. Still in the area of binaural rendering are the two contributions to this special issue, the first of which is by L. Wang et al., which addresses the long-debated problem of cross-talk cancellation. This paper is followed by that of M. Cobos et al., which proposes a method that allows us to avoid using a dummy head in binaural recording sessions.

This special issue also includes two papers that deal with high-level processing of musical content, which can be used for a variety of applications, from music information retrieval to digital audio effects. The former, by A. Barbancho et al., is concerned with piano chords detection based on parallel interference cancellation methods. The latter, by Itoyama et al., and Okuno, tackles a query-by-example technique based on source separation and remixing.

Augusto Sarti
Udo Zoelzer
Xavier Serra
Mark Sandler
Simon Godsill

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 392782, 12 pages
doi:10.1155/2010/392782

Research Article

Advances in Modal Analysis Using a Robust and Multiscale Method

Cécile Picard,[1] Christian Frisson,[2] François Faure,[3] George Drettakis,[1] and Paul G. Kry[4]

[1] *REVES/INRIA, BP 93, 06902 Sophia Antipolis, France*
[2] *TELE Lab, Université catholique de Louvain (UCL), 1348 Louvain-la-Neuve, Belgium*
[3] *EVASION/INRIA/LJK, Rhône-Alpes Grenoble, France*
[4] *SOCS, McGill University, Montreal, Canada H3A 2A7*

Correspondence should be addressed to Cécile Picard, cecile.picard@sophia.inria.fr

Received 1 March 2010; Revised 29 June 2010; Accepted 18 October 2010

Academic Editor: Xavier Serra

This paper presents a new approach to modal synthesis for rendering sounds of virtual objects. We propose a generic method that preserves sound variety across the surface of an object at different scales of resolution and for a variety of complex geometries. The technique performs automatic voxelization of a surface model and automatic tuning of the parameters of hexahedral finite elements, based on the distribution of material in each cell. The voxelization is performed using a sparse regular grid embedding of the object, which permits the construction of plausible lower resolution approximations of the modal model. We can compute the audible impulse response of a variety of objects. Our solution is robust and can handle nonmanifold geometries that include both volumetric and surface parts. We present a system which allows us to manipulate and tune sounding objects in an appropriate way for games, training simulations, and other interactive virtual environments.

1. Introduction

Our goal is to realistically model sounding objects for animated realtime virtual environments. To achieve this, we propose a robust and flexible modal analysis approach that efficiently extracts modal parameters for plausible sound synthesis while also focusing on efficient memory usage.

Modal synthesis models the sound of an object as a combination of damped sinusoids, each of which oscillates independently of the others. This approach is only accurate for sounds produced by linear phenomena, but can compute these sounds in realtime. It requires the computation of a partial eigenvalue decomposition of the system matrices of the sounding object, which can be expensive for large complex systems. For this reason, modal analysis is performed in a preprocessing step. The eigenvalues and eigenvectors strongly depend on the geometry, material and scale of the sounding object. In general, complex sounding objects, that is, with detailed geometries, require a large set of eigenvalues in order to preserve the *sound map*, that is, the changes in sound across the surface of the sounding object. This processing step can be subject to robustness problems. This is even more the case for nonmanifold geometries, that is, geometries where one edge is shared by more than two faces. Finally, available approaches manage memory usage in realtime by only pruning part of modal parameters according to the characteristics of the virtual scene (e.g., foreground versus background), without specific consideration regarding the objects' sound modelling. Additional flexibility in the modal analysis itself is thus needed.

We propose a new approach to efficiently extract modal parameters for any given geometry, overcoming many of the aforementioned limitations. Our method employs bounding voxels of a given shape at arbitrary resolution for hexahedral finite elements. The advantages of this technique are the automatic voxelization of a surface model and the automatic tuning of the finite element method (FEM) parameters based on the distribution of material in each cell. A particular advantage of this approach is that we can easily deal with nonmanifold geometry which includes both volumetric and surface parts (see Section 5). These kinds of geometries cannot be processed with traditional approaches which use

a tetrahedralization of the model (e.g., [1]). Likewise, even with solid watertight geometries, complex details often lead to poorly shaped tetrahedra and numerical instabilities; by contrast, our approach does not suffer from this problem. Our specific contribution is the application of the multiresolution hexahedral embedding technique to modal analysis for sound synthesis. Most importantly, our solution preserves variety in what we call the *sound map*.

The remainder of this paper is organized as follows. Related work is presented in Section 2. Our method is then explained in Section 3. A validation is presented in Section 4. Robustness and multiscale results are discussed in Section 5, then realtime experimentation is presented in Section 6. We finally conclude in Section 7.

2. Background

2.1. Related Work. The traditional approach to creating soundtracks for interactive physically based animations is to directly playback prerecorded samples, for instance, synchronized with the contacts reported from a rigid-body simulation. Due to memory constraints, the number of samples is limited, leading to repetitive audio. Moreover, matching sampled sounds to interactive animation is difficult and often leads to discrepancies between the simulated visuals and their accompanying soundtrack. Finally, this method requires each specific contact interaction to be associated with a corresponding prerecorded sound, resulting in a time-consuming authoring process.

Work by Adrien [2] describes how effective digital sound synthesis can be used to reconstruct the richness of natural sounds. There has been much work in computer music [3–5] and computer graphics [1, 6, 7] exploring methods for generating sound based on physical simulation. Most approaches target sounds emitted by vibrating solids. Physically based sounds require significantly more computation power than recorded sounds. Thus, brute-force sound simulation cannot be used for realtime sound synthesis. For interactive simulations, a widely used solution is to apply vibrational parameters obtained through modal analysis. Modal data can be obtained from simulations [1, 7] or extracted from recorded sounds of real objects [6]. The technique presented in this paper is more closely related to the work of O'Brien et al. [1], which extends modal analysis to objects that are neither simple shapes nor available to be measured.

The computation time required by current methods to preprocess the modal analysis prevents it from being used for realtime rendering. As an example, the actual cost of computing the partial eigenvalue decomposition using a tetrahedralization in the case of a bowl with 274 vertices and generating 2426 tetrahedra is 5 minutes with a 2.26 GHz Intel Core Duo. Work of Bruyns-Maxwell and Bindel [8] address interactive sound synthesis and how the change of the shape of a finite element model affects the sound emission. They highlight that it is possible to avoid the recomputation of the synthesis parameters only for moderate changes. There has been much work in controlling the computational expense of modal synthesis, allowing the simultaneous handling of a large variety of sounding objects [9, 10]. However, to be even more efficient, flexibility should be included in the design of the model itself, in order to control the processing. Thus, modal synthesis should be further developed in terms of parametric control properties. Our technique tackles computational efficiency by proposing a multiscale resolution approach of modal analysis, managing the amount of modal data according to memory requirements.

The use of physics engines is becoming much more widespread for animated interactive virtual environments. The study from Menzies [11] address the pertinence of physical audio within physical computer game environment. He develops a library whose technical aspects are based on practical requirements and points out that the interface between physics engines and audio has often been one of the obstacles for the adoption of physically based sound synthesis in simulations. O'Brien et al. [12] employed finite elements simulations for generating both animated videos and audio. However, the method requires large amounts of computation, and cannot be used for realtime manipulation.

2.2. Modal Synthesis. Modal sound synthesis is a physically based approach for modelling the audible vibration modes of an object. As any kind of additive synthesis, it consists of describing a source as the sum of many components [13]. More specifically, the source is viewed as a bank of damped harmonic oscillators which are excited by an external stimulus and the modal model is represented with the vector of the modal frequencies, the vector of the decay rates and the matrix of the gains for each mode at different locations on the surface of the object. The frequencies and dampings of the oscillators are governed by the geometry and material properties of the object, whereas the coupling gains of the modes are determined by the mode shapes and are dependent on the contact location on the object [6].

Modes are computed through an analysis of the governing equations of motion of the sounding system. The natural frequencies are determined assuming the dynamic response of the unloaded structure, with the equation of motion. A system of n degrees of freedom is governed by a set of n coupled ordinary differential equations of second order. In modal analysis, the deformation of the system is assumed to be a linear combination of normal modes, uncoupling the equations of motion. The solution for object vibration can be thus easily computed. To decouple the damped system into single degree-of freedom oscillators, Rayleigh damping is generally assumed (see, for instance, [14]).

The response of a system is usually governed by a relatively small part of the modes, which makes modal superposition a particularly suitable method for computing the vibration response. Thus, if the structural response is characterized by k modes, only k equations need to be solved. Finally, the initial computational expense in calculating the modes and frequencies is largely offset by the savings obtained in the calculation of the response.

Modal synthesis is valid only for linear problems, that is, simulations with small displacements, linear elastic materials, and no contact conditions. If the simulation presents

nonlinearities, significant changes in the natural frequencies may appear during the analysis. In this case, direct integration of the dynamic equation of equilibrium is needed, which requires much more computational effort. For our approach, the calculations for modal parameters are similar to the ones presented in the paper of O'Brien et al. [1].

3. Method

In the case of small elastic deformations, rigid motion of an object does not interact with the objects's vibrations. On the other hand, we assume that small-amplitude elastic deformations will not significantly affect the rigid-body collisions between objects. For these reasons, the rigid-body behavior of the objects can be modeled in the same way as animation without audio generation.

3.1. Deformation Model. In most approaches, the deformation of the sounding object typically need to be simulated. Instead of directly applying classical mechanics to the continuous system, suitable discrete approximations of the object geometry can be performed, making the problem more manageable for mathematical analysis. A variety of methods could be used, including particle systems [3, 7] that decompose the structure into small pair-like elements for solving the mechanics equations, or Boundary Element Method (BEM) that computes the equations on the surface (boundary) of the elastic body instead of on its volume (interior), allowing reflections and diffractions to be modeled [15, 16]. The Finite Element Method (FEM) is commonly used to perform modal analysis, which in general gives satisfactory results. Similar to particle systems, FEM discretizes the actual geometry of the structure using a collection of finite elements. Each finite element represents a discrete portion of the physical structure and the finite elements are joined by shared nodes. The collection of nodes and finite elements is called a mesh. The tetrahedral finite element method has been used to apply classical mechanics [1]. However, tetrahedral meshes are computationally expensive for complex geometries, and can be difficult to tune. As an example, in the tetrahedral mesh generator *Tetgen* (http://tetgen.berlios.de/), the mesh element quality criterion is based on the minimum radius-edge ratio, which limits the ratio between the radius of the circumsphere of the tetrahedron and the shortest edge length. Based on this observation, we choose a finite elements approach whose volume mesh does not exactly fit the object.

We use the method of Nesme et al. [17] to model the small linear deformations that are necessary for sound rendering. In this approach, the object is embedded in a regular grid where each cell is a finite element, contrary to traditional FEM models where the elements try to match the object geometry as finely as possible. Tuning the grid resolution allows us to easily trade off accuracy for speed. The object is embedded in the cells using barycentric coordinates. Though the geometry of the mesh is quite different from the object geometry, the mechanical properties (mass and stiffness) of the cells match as closely as possible the spatial distribution and the parameters of material. The technique can be summarized as follows. An automatic high-resolution voxelization of the geometric object is first built. The voxelization initially concerns the surface of the geometric model, while the interior is automatically filled when the geometry represents a solid object. The voxels are then recursively merged (8 to 1) up to the desired coarser mechanical resolution. The merged voxels are used as hexahedral (boxes with the same shape ratio as the fine voxels) finite elements embedding the detailed geometric shape. The voxels are usually cubes but they may have different sizes in the three directions. At each step of the coarsification, the stiffness and mass matrices of a coarse element are computed based on the eight child element matrices. Mass and stiffness are thus deduced from a fine grid to a coarser one, where the finest depth is considered close enough to the surface, and the procedure can be described as a two-level structure, that is, from fine to coarse grid. The stiffness and mass matrices are computed bottom-up using the following equation:

$$\mathbf{K}_{\text{parent}} = \sum_{i=0}^{7} \mathbf{L}_i^T \mathbf{K}_i \mathbf{L}_i, \tag{1}$$

where $\mathbf{K}$ is the matrix of the parent node, the $\mathbf{K}_i$ are the matrices in the child nodes, and the $\mathbf{L}_i$ are the interpolation matrices of the child cell vertices within the parent cell. Since empty children have null $\mathbf{K}_i$ matrices, the fill rate is automatically taken into account, as well as the spatial distribution of the material through the $\mathbf{L}_i$ matrices. As a result, full cells are heavier and stiffer than partially empty cells, and the matrices not only encode the fill rate but also the distribution of the material within each cell. With this method, we can handle objects with geometries that simultaneously include volumetric and surface parts; thin or flat features will occupy voxels and will thus result in the creation of mechanical elements that robustly approximate their mechanical behavior (see Section 5.1).

3.2. Modal Analysis. The method for FEM model [17] is adapted from realtime deformation to modal analysis. In particular, the modal parameters are extracted in a preprocessing step by solving the equation of motion for small linear deformations. We first compute the global mass and global stiffness matrices for the object by assembling the element matrices. In the case of three-dimensional objects, global matrices will have a dimension of $3m \times 3m$ where m is the number of nodes in the finite element mesh. Each entry in each of the 24×24 element matrices for a cell is accumulated into the corresponding entry of the global matrix. Because each node in the hexahedral mesh shares an element with only a small number of the other nodes, the global matrices will be sparse. If we assume the displacements are small, the discretized system is described on a mechanical level by the Newton second law

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{f}, \tag{2}$$

where $\mathbf{d}$ is the vector of node displacements, and a derivative with respect to time is indicated by an overdot. $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are, respectively, the system's mass, damping and

(1) Compute mass and stiffness at desired mechanical level
(2) Assemble the mass and the stiffness matrices
(3) Modal analysis: solve the eigenproblem
(4) Store eigenvalues and eigenvectors for sound synthesis

Algorithm 1: Algorithm for modal parameters extraction.

stiffness matrices, and **f** represents external forces, such as impact forces that will produce audible vibrations. Assuming Rayleigh damping, that is, $\mathbf{C} = \alpha_1\mathbf{K} + \alpha_2\mathbf{M}$ with some α_1 and α_2, we can solve the eigenproblem of the decoupled system leading to the n eigenvalues and the $n \times m$ matrix of eigenvectors, with n the number of degrees of freedom and m the number of nodes in the mesh. The sparseness of **M** and **K** matrices allows the use of sparse matrix algorithms for the eigen decomposition. We refer the reader to Appendices A and B for more details on the calculation.

Let λ_i be the ith eigenvalue and ϕ_i its corresponding eigenvector. The eigenvector, also known as the mode shape, is the deformed shape of the structure as it vibrates in the ith mode. The natural frequencies and mode shapes of a structure are used to characterize its dynamic response to loads in the linear regime. The deformation of the structure is then calculated from a combination of the mode shapes of the structure using the modal superposition technique. The vector of displacements of the model, **u**, is defined as:

$$\mathbf{u} = \sum \beta_i \phi_i, \tag{3}$$

where β_i is the scale factor for mode ϕ_i. The eigenvalue for each mode is determined by the ratio of the mode's elastic stiffness to the mode's mass. For each eigen decomposition, there will be six zero eigenvalues that correspond to the six rigid-body modes, that is, modes that do not generate any elastic forces.

Our preprocessing step that performs modal analysis can be summarized as in Algorithm 1.

Our model approximates the motion of the embedded mesh vertices. That is, the visual model with detailed geometry does not match the mechanical model on which the modal analysis is performed. The motion of the embedding uses a trilinear interpolation of the mechanical degrees of freedom, so we can nevertheless compute the motion of any point on the surface given the mode shapes.

3.3. Sound Generation. In essence, efficiency of modal analysis relies on neglecting the spatial dynamics and modelling the actual physical system by a corresponding generalized mass-spring system which has the same spectral response. The activation of this model depends on where the object is hit. If we hit the object at a vibration node of a mode, then that mode will not vibrate, but others will. This is what we refer to as the *sound map*, which could also be called a sound excitation map as it indicates how the different modes are excited when the object is struck at different locations.

From the eigenvalues and the matrix of eigenvectors, we are able to deduce the modal parameters for sound synthesis. Let λ_i be the ith eigenvalue and ω_i its square root. The absolute value of the imaginary part of ω_i gives the natural frequency (in radians/second) of the ith mode of the structure, whereas the real part of ω_i gives the mode's decay rate. The mode's gain is deduced from the eigenvectors matrix and depends on the excitation location. We refer the reader to the Appendices A and B for more details on modal superposition.

The sound resulting from an impact on a specific location j on the surface is calculated as a sum of n damped oscillators:

$$s_j(t) = \sum_{i=1}^{n} a_{ij} \sin(2\pi f_i t) e^{-d_i t}, \tag{4}$$

where f_i, d_i, and a_{ij} are, respectively, the frequency, the decay rate and the gain of the mode i at point j in the sound map. An object characterized with m mesh nodes and n degrees-of freedom is described with the vectors of frequencies and decay rates of dimension n, and the matrix of gains of dimension $n \times m$.

3.4. Implementation. Our deformation model implementation uses the *SOFA Framework* (http://www.sofa-framework.org/) for small elastic deformations. SOFA is an open-source C++ library for physical simulation and can be used as an external library in another program, or using one of the associated GUI applications. This choice was motivated by the ease with which it could be extended for our purpose.

Regarding sound generation, we synthesize the sounds via a reson filter (see, for example, Van den Doel et al. [6]). This choice is made based on the effectiveness for realtime audio processing. Sound radiation amplitudes of each mode is also estimated with a far-field radiation model ([15, Equation (15)]). As the motions of objects are computed with modal analysis, surfaces can be easily analyzed to determine how the motions induce acoustic pressure waves in the surrounding medium. However, we decide to focus our study on effective modal synthesis. Finally, our approach does not consider contact-position-dependent damping or changes in boundary constraints, as might happen during moments of excitation. Instead we use a uniform damping value for the sounding object.

4. Validation of the Model

4.1. A Metal Cube. In order to globally validate our method for modal analysis, we study the sound emitted when impacting a cube in metal. Due to its symmetry, the cube should sound the same when struck at any of the eight corners, with an excitation force whose direction is the same to the face (see Appendices A and B for more details on the force amplitude vector). We use a force normal to the face cube in order to guarantee the maximum energy in all excited modes. The sound emitted should also be similar when hitting with perpendicular forces that are both normal to one pair of the cube faces.

We suppose the cube is made of steel with Young's modulus 21×10^{10} Pa, Poisson ratio 0.33, and density

7850 kg/m^3. The Raleigh coefficients for stiffness and mass are set to 1×10^{-7} and 0, respectively. The use of a constant damping ratio is a simplification that still produces good results. The cube model has edges which are 1 meter long. A Dirac is chosen for the excitation force. In this case, no radiation properties are considered.

In this example, a $3 \times 3 \times 3$ grid of hexahedral finite elements is used, leading to 192 modes. However, to adapt the stiffness of a cell according to its content, the mesh is refined more precisely than desired for the animation. The information is propagated from fine cells to coarser cells. For this example, the elements of the $3 \times 3 \times 3$ cells coarse grid resolution approximates mechanical properties propagated from a fine grid of $6 \times 6 \times 6$ cells and 216 elements (see Section 3.1 for more details on the two-level structure).

We observe in Figure 1 that the resulting sounds when impacting on different corners of the cube are identical. Also, this is true when exciting with perpendicular forces that are normal to cube faces. This shows that our model respects the symmetry of objects, as expected.

4.2. Position-Dependent Sound Rendering. To properly render impact sounds of an object, the method must preserve the sound variety when hitting the surface at different locations. We consider a metal bowl, modeled by a triangle mesh with 274 vertices, shown in Figure 2.

The material of the bowl is aluminium, with the parameters 69×10^9 Pa for Young's modulus, 0.33 for Poisson ratio, and 2700 kg/m^3 for the density. The Rayleigh damping parameters for stiffness and mass are set to 3×10^{-6} and 0.01, respectively. The bowl has a width of 1 meter. No radiation properties are considered; our study focuses specifically on modal synthesis.

We compare our approach to modal analysis performed first using tetrahedralization with *Tetgen* (http://tetgen.berlios.de/) with 822 modes. Our method uses hexahedral finite elements and is applied with a grid of $6 \times 6 \times 6$ cells, leading to 891 modes. For this example, the elements of the $6 \times 6 \times 6$ cells coarse grid resolution approximates mechanical properties propagated from a fine grid of $12 \times 12 \times 12$ cells.

We first compare the extracted modes from both methods. We observe that the ratio between frequencies and decays is the same for both methods. We then compare the synthesized sounds from both methods. We take 3 different locations, that is, top, side and bottom, on the surface of the object where the object is impacted, see Figure 2. The excitation force is modeled as a Dirac, such as a regular impact. The frequency content of the sound resulting from impact at the 3 locations on the surface is shown in Figure 3.

Each power spectrum is normalized with the maximum amplitude in order to factor out the magnitude of the impact. The eigenvalues that correspond to vibration modes will be nonzero, but for each free body in the system there will be six zero eigenvalues for the body's six rigid-body freedoms. Only the modes with nonzero eigenvalue are kept. Thus, 816 modes are finally used for sound rendering with the tetrahedralization method and 885 with our hexahedral FEM method.

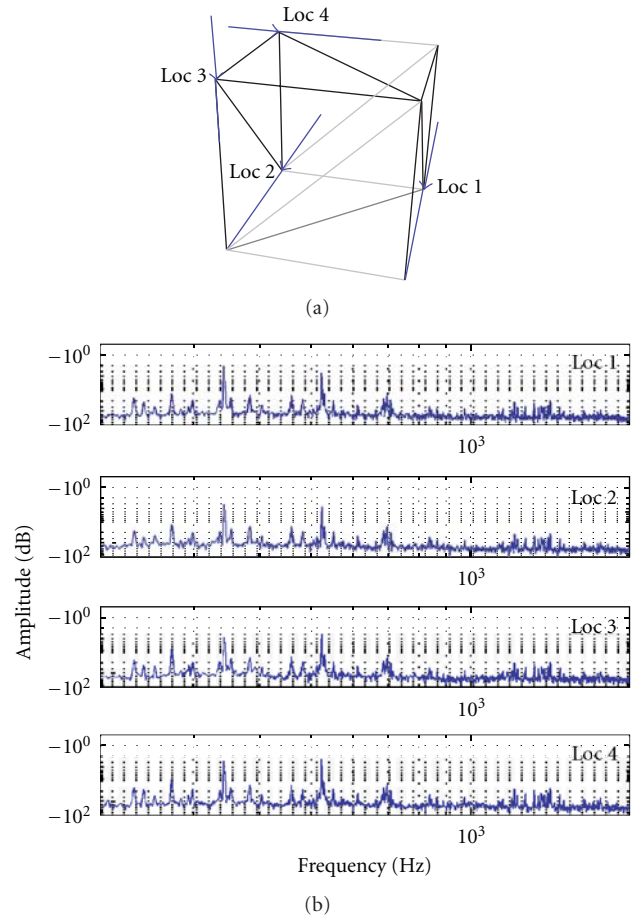


Figure 1: A sounding metal cube: sound synthesis is performed for excitation on 4 different corners and forces normal to one pair of cube faces (a); the power spectrum of the emitted sounds is given (b).

We provide with the sounds synthesized with the tetrahedral FEM and the hexadedral FEM approaches (see additional material (http://www-sop.inria.fr/members/Cecile.Picard/Material/AdditionalMaterialEurasip.zip)). Figure 3 highlights the similarities in the main part of the frequency content. The difference when impacting at the bottom (location 3) of the object is due to the difference in distribution of modes and we believe this is due to the size of the finite elements used in our method. However, we notice in listening to the synthesized sounds that those generated by our method are comparable to those created with the standard tetrahedralization.

5. Robustness and Multiscale Results

The number of finite elements determine the dimension of the system to solve. To avoid this expense, we provide a method that greatly simplifies the modal parameter extraction even for nonmanifold geometries. An important subclass of nonmanifold models are objects that include both volumetric and surface parts. Our technique consists of using multiresolution hexahedral embeddings.

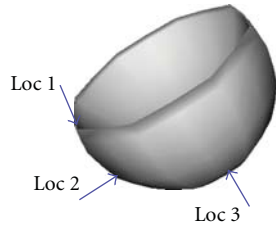


Figure 2: A sounding metal bowl: sound synthesis is performed for excitation on 3 specific locations on the surface.

5.1. Robustness. Most approaches for tetrahedral mesh generation have limitations. In particular, an important requirement imposed by the application of deformable FEM is that tetrahedra must have appropriate shapes, for instance, not too flat or sharp. By far the most popular of the tetrahedral meshing techniques are those utilizing the Delaunay criterion [18]. When the Delaunay criterion is not satisfied, modal analysis using standard tetrahedralization is impossible. In comparison with tetrahedralization methods, our technique can handle complex geometries and adequately performs modal analysis. Figures 4 and 5 give an example of sound modelling on a problematic geometry for tetrahedralization because of the presence of very thin parts, specifically the blades that protrude from either side.

We suppose the object is made of aluminum (see Section 4.2 for the material parameters). The object has a height of 1 meter. We apply a coarse grid of $7 \times 7 \times 7$ cells for modal analysis. The coarse level encloses the mechanical properties of a fine grid of $14 \times 14 \times 14$ cells (see Section 3.1 for more details on the two-level structure). In this example, sound radiation amplitudes of each mode are also estimated with a far-field radiation model [15, Equation (15)]. Figure 5 shows the power spectrum of the sounds resulting from impacts, modeled as a Dirac, on 6 different locations. Each power spectrum is normalized with the maximum amplitude of the spectrum in order to factor out the magnitude of the impact.

We provide with the sounds resulting when hitting on the 6 different locations (see additional material, link referred in Section 4.2). Figure 5 shows the variation of impact sounds at different surface locations due to the sound map since the different modes have varying amplitude depending on the location of excitation. The frequency content is related to the distribution of mass and stiffness along the surface and more precisely to the ratio between stiffness and mass. The similarities in the resulting sounds when hitting on location 1 and location 3 are due to the similarities of the local geometry. However, the stiffness at location 3 is smaller, allowing more resonance when being struck which explains the predominant peak in the corresponding power spectrum. When hitting the body of the object at location 2, the stiffness is locally smaller in comparison to locations 1 and 3, leading to a larger amount of low-frequency content. Also, it is interesting to examine the quality of the sound rendered when hitting the wings (locations 4, 5, and 6). Because wings are thin and light in comparison to the rest of the object, the higher frequencies are more pronounced. Finally, impacts on locations 2 and 4 gives comparable sounds since the impact locations are close on the body of the object.

5.2. A Multi-scale Approach. To study the influence of the number of hexahedral finite elements on sound rendering, we model a sounding object with different resolutions of hexahedral finite elements. We have created a squirrel model with 999 vertices which we use as our test sounding object. The squirrel model has a height of 1 meter. Its material is pine wood, which has parameters 12×10^9 Pa for Young's modulus, 0.3 for Poisson ratio, and $750\,\text{kg/m}^3$ for the volumetric mass. Rayleigh damping parameters for stiffness and mass are set to 8×10^{-6} and 50, respectively.

Sound synthesis is performed for 3 different locations of excitation, see Figure 6 (top left). The coarse grid resolution for finite elements is set to $2 \times 2 \times 2$, $3 \times 3 \times 3$, $5 \times 5 \times 5$, and $7 \times 7 \times 7$. In this example, each grid uses mass and stiffness computed as described in Section 3.1 from a resolution 4 times finer; that is, the model with resolution $2 \times 2 \times 2$ has properties computed with a grid of $8 \times 8 \times 8$.

We provide with the sounds synthesized with the different grid resolutions for finite elements and for the 3 different locations of excitation (see additional material, link referred in Section 4.2). Results show that the frequency content of sounds depend on the location of excitation and on the resolution of the hexahedral finite elements. The higher resolution models have a wider range of frequencies because of the supplementary degrees of freedom. We also observe a frequency shift as the FEM resolution increases. Note that a $2 \times 2 \times 2$ grid represents an extremely coarse embedding, and consequently it is not surprising that the frequency content is different at higher resolution. Nevertheless, there are still some strong similarities at the dominant frequencies. Above all, a desirable feature is the convergence of frequency content as the resolution of the model increases. While additional psychoacoustic experiments with objective spectral distortion measures would be necessary to validate this result, when listening to the results, the sound quality for this model at a grid of $5 \times 5 \times 5$ may produce a convincing sound rendering for the human ear. Figure 6 suggests that higher resolutions are necessary before convergence can be clearly observed in the frequency content. Finally, we note that the grid resolution required for acceptable precision in the sound rendering depends on the geometry of the simulated object.

5.3. Discussion. The *sound map* is influenced by the resolution of the hexahedral finite elements. This is related to the way stiffnesses and masses of different elements are altered based on their contents. As a consequence, a $2 \times 2 \times 2$ hexahedral FEM resolution would show much less expressive variation than higher FEM resolution (we refer the reader to the records provided in the additional material, link referred in Section 4.2). One approach to improving this would be to use better approximations of the mass and stiffness of coarse elements [19].

Modelling numerous complex sounding objects can rapidly become prohibitively expensive for realtime

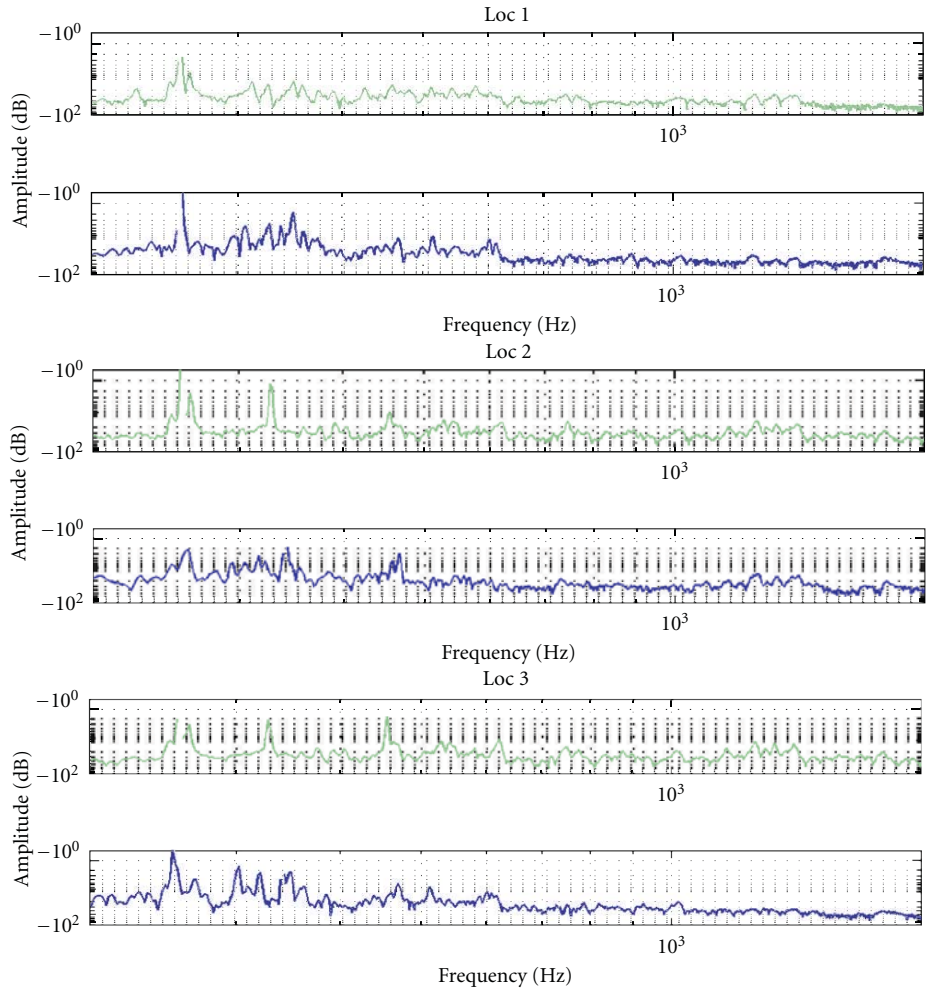


Figure 3: Sound synthesis with a modal approach using classical tetrahedralization with 822 modes (green) and our method with a $6 \times 6 \times 6$ hexahedral FEM resolution, leading to 891 modes (blue): power spectrum of the sounds emitted when impacting at the 3 different locations shown in Figure 2.

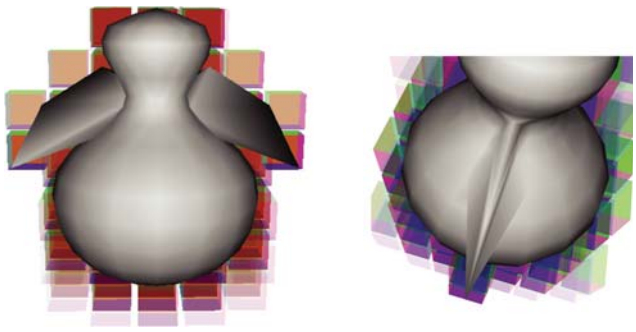


Figure 4: An example of a complex geometry that can be handled with our method. The thin blade causes problems with traditional tetrahedralization methods.

rendering due to the large set of modal data that has to be handled. Nevertheless, based on the quality of the resulting sounds obtained with our method, and given that increased resolution for the finite elements implies higher memory and computational requirements for modal data, the FEM resolution can be adapted to the number of sounding objects in the virtual scene.

Table 1 gives the computation times and the memory usage of the modal data, that is, frequencies, decay rates and gains, when computing the modal analysis with different FEM resolution on the squirrel model. In this example, the finer grid resolution is two levels up to the one of coarse grid, that is, a coarse grid of $2 \times 2 \times 2$ cells has a fine level of $8 \times 8 \times 8$ cells with 337 degrees of freedom (3954 for $5 \times 5 \times 5$). These are computation times of an unoptimized implementation on a 2.26 GHz Intel Core Duo. We highlight the $5 \times 5 \times 5$ cells resolution since the results indicate that this resolution may be sufficient to properly render the sound quality of the object (see Section 5.2). These results could be improved by reformulating the computations in order to be supported by graphics processing units (GPU).

Despite the fact that audio is considered a very important aspect in virtual environments, it is still considered to be of lower importance than graphics. We believe that physically modeled audio brings a significant added value in terms

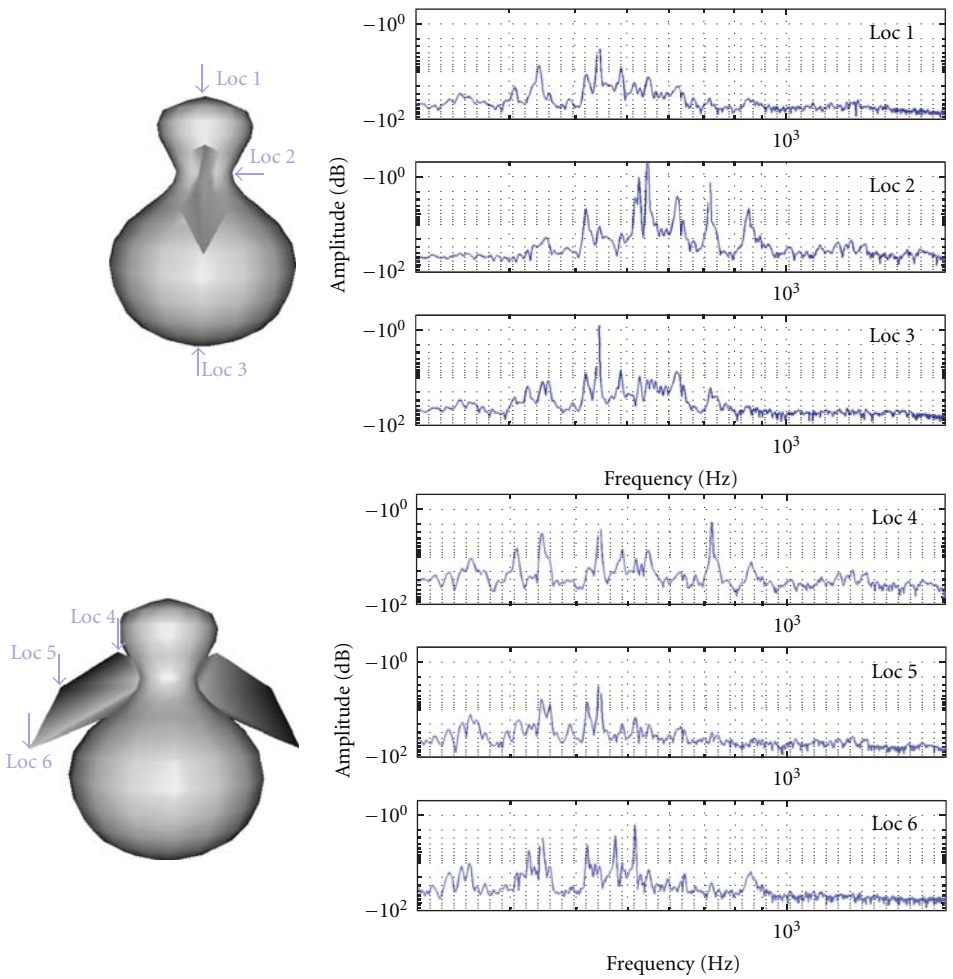


Figure 5: The power spectrum of the sounds resulting from impacts at the 3 different locations on the body of the object (top) and on the 3 different locations on the wing (bottom). Note that the audible response is different based on where the object is hit.

Table 1: Computation times in seconds and memory usage in megabytes for different grid resolutions. Computation times are given for the different steps of the calculation: discretization and computation of mass and stiffness matrices (T1), eigenvalues extraction (T2), and gains computation (T3).

Grid Res. (cells)	# Modes	T1 (s)	T2 (s)	T3 (s)	Total (s)	MEM (MB)
$7 \times 7 \times 7$	1191	1.81	16.06	3.99	21.89	9.3
$6 \times 6 \times 6$	846	0.89	5.78	2.39	9.06	6.8
$5 \times 5 \times 5$	**579**	**0.43**	**2.07**	**0.97**	**3.47**	**4.7**
$4 \times 4 \times 4$	363	0.24	0.61	0.59	2.88	2.9
$3 \times 3 \times 3$	192	0.05	0.14	0.16	0.35	1.6
$2 \times 2 \times 2$	81	0.01	0.03	0.01	0.05	0.69

of realism and the increased sense of immersion. Our method is built on a physically based animation engine, the *SOFA Framework*. As a consequence, problems of coherence between physics simulation and audio are avoided by using exactly the same model for simulation and sound modelling. The sound can be processed in realtime knowing the modal parameters of the sounding object.

6. Experimenting with the Modal Sounds in Realtime

To apply excitation signals in realtime to the simulated sounding objects, we implemented an object, or data processing block, for Pure Data and Max/MSP, two similar visual programming modular environments for dataflow processing. We used the *flext* library (http://puredata.info/Members/thomas/flext/) (API for object development common to both environments), and the C/C++ code for modal synthesis of bell sounds from van den Doel and Pai [20]. The object in use on a Pure Data patch is illustrated in Figure 7). We provide the user two different ways for the user to interact with the model. The user can either choose a specific mesh vertex number of the geometry model (represented in red in the figure), or can choose a specific location (in green) where the nearest vertex is deduced by interpolation.

One advantage of the method is to give the possibility to control the parameters of the sounding model in order

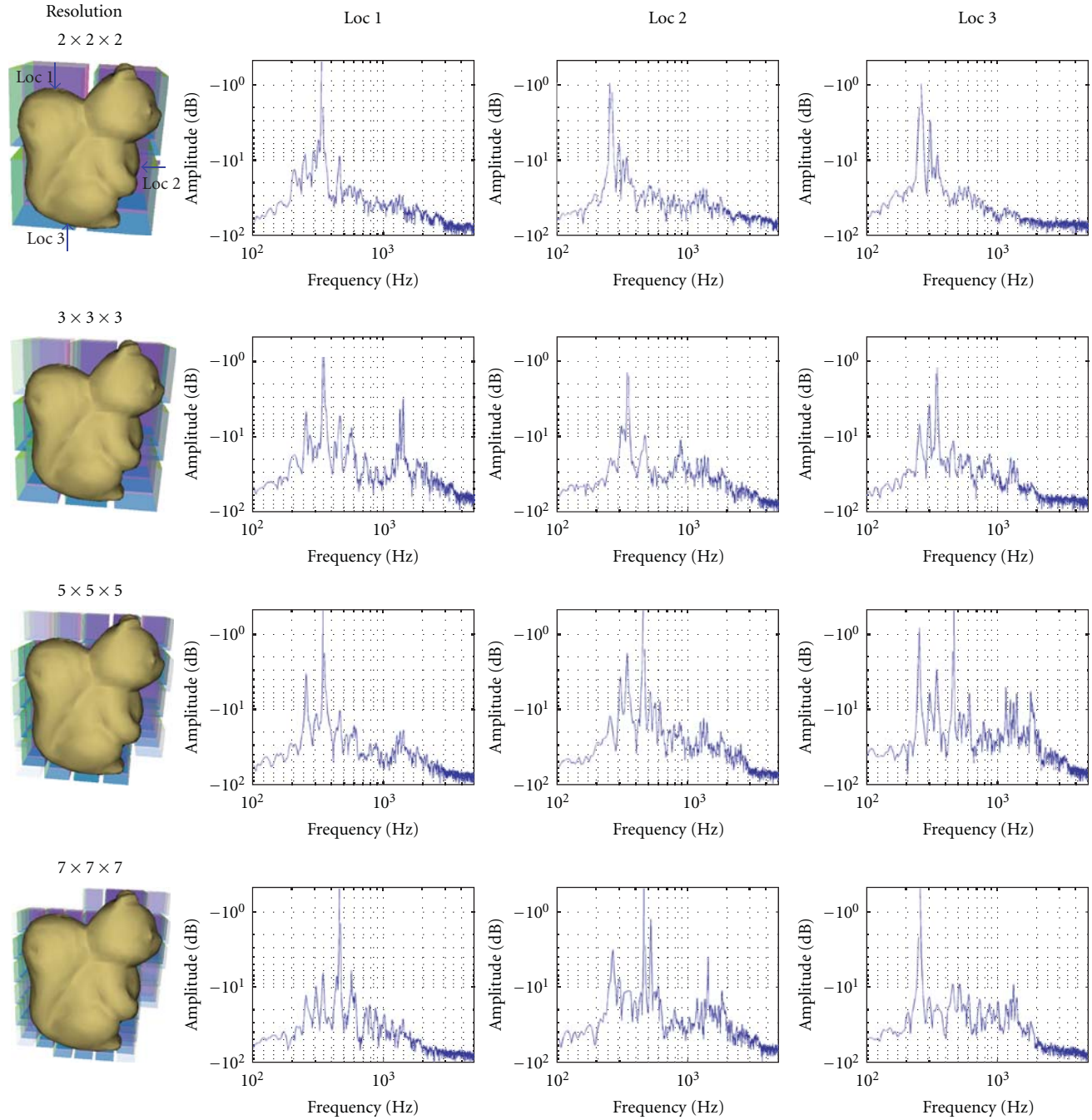


FIGURE 6: A squirrel in pine wood is sounding when struck at 3 different locations (from left to right). Frequency content of the resulting sounds with 4 different resolutions for the hexahedral finite elements: (from top to bottom), $2 \times 2 \times 2$, $3 \times 3 \times 3$, $5 \times 5 \times 5$, $7 \times 7 \times 7$ cells.

to tune the resulting sounds for the desired effect. For instance, the size of the geometry can be modified as different dimensions could be preferred for rendering sounds in a particular scenario. The mesh geometry is loaded in *Alias\Wavefront *.obj* format, and we use Blender to apply geometrical transformations in order to test how it affects the rendering of the resulting sounds.

As our sound model consists of an excitation and a resonator, interesting sounds can be easily obtained by convolving modal sounds with user-defined excitations. The excitation which supplies the energy to the sound system contributes to a great extent to the fine details of the resulting sounds. Excitation signals may be produced by various ways: loading recorded sound samples, using realtime signals coming from live soundcard inputs, connecting the output of other audio applications with Pure Data through a sound server.

This interface can be viewed as a preliminary prototyping tool for sound design. Indeed, by experimenting sounds with predefined objects and interactions types, the parameters of

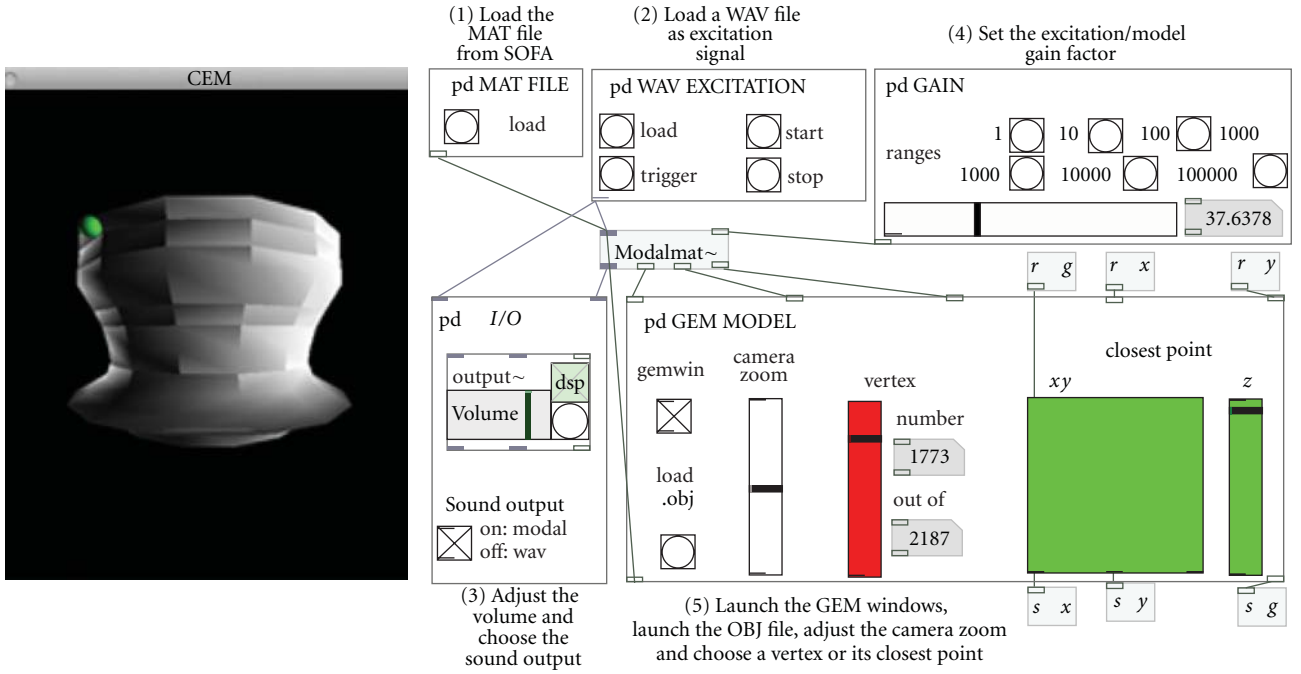


Figure 7: Interface for sound design. After having loaded the modal data and the corresponding mesh geometry, the user can experiment the modal sounds when exciting the object surface at different locations. Excitation signals may be loaded as recorded sound samples or realtime tracked from live soundcard inputs.

sounding objects can easily chosen in order to convey specific sensations in games. Our approach offers a great extent of control regarding the possibilities of sound modification, towards a wide audience since its implementation is cross-platform and open source. In [21], Bruyns proposed an AudioUnit plugin, that is unfortunately no longer available, for modal synthesis of arbitrarily shaped objects, where materials could be changed based on interpolation between precalculated variations on the model. Lately, Menzies has introduced VFoley in [22], an opensource environment for modal synthesis of 3D scenes, with consequent options on parameterization (particularly with many collision and surface models), but tied to physically plausible sounds as opposed to physically-inspired sounds. This is shown in the movie provided as additional material (see link referred in Section 4.2).

7. Conclusion

We propose a new approach to modal analysis using automatic voxelization of a surface model and computation of the finite elements parameters, based on the distribution of material in each cell. Our goal is to perform sound rendering in the context of an animated realtime virtual environment, which has specific requirements, such as realtime processing and efficient memory usage.

For simple cases, our method gives results similar to traditional modal analysis with tetrahedralization. For more complex cases, our approach provides convincing results. In particular, sound variety along the object surface, the *sound map*, is well preserved. Our technique can handle complex nonmanifold geometries that include both volumetric and surface parts, which cannot be handled by previous techniques. We are thus able to compute the audio response of numerous and diverse sounding objects, such as those used in games, training simulations, and other interactive virtual environments. Our solution allows a multiscale solution because the number of hexahedral finite elements only loosely depends on the geometry of the sounding object. Finally, since our method is built on a physics animation engine, the *SOFA Framework*, problems of coherence between simulation and audio can be easily addressed, which is of great interest in the context of interactive environment.

In addition, due to the fast computation time, we are hopeful that realtime modal analysis will soon be possible on the fly, with sound results that are approximate but still realistic for virtual environments. For this purpose, psychoacoustic experiments should be conducted to determine the resolution level for acceptable quality of the sound rendering.

Appendices

These appendices give the mathematical background behind modal superposition for discrete systems with proportional damping. To apply modal superposition, we assume the steady state situation, that is, the sustained part of the impulse response of an object being struck. Indeed, the early part, which is of very short duration, contains many frequencies and is consequently not well described by a discrete set of frequencies. Modal superposition uses the

Finite Element Method (FEM) and determine the impulse response of vibrating objects by means of a superposition of eigenmodes.

A. Derivation of the Equations

We first consider the undamped system; its equation of motion is expressed by

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}, \tag{A.1}$$

where $\mathbf{M}$ and $\mathbf{K}$ are, respectively, the mass and stiffness matrices of the discrete system. The mass matrix is typically a diagonal matrix, its main diagonal being populated with elements whose value is the mass assumed in each degree of freedom (DOF). The stiffness matrix is symmetric (often a sparse matrix, that is, only a band of elements around the main diagonal is populated and the other elements are zero). In finite elements, these matrices are assembled based on the element geometry and properties.

Since the study is in the frequency domain, the displacement vector $\mathbf{x}$ and the force vector $\mathbf{f}$ are based on harmonic components, that is, $\mathbf{x} = \mathbf{X}e^{j\omega t}$, $\dot{\mathbf{x}} = j\omega\mathbf{X}e^{j\omega t}$, $\ddot{\mathbf{x}} = -\omega^2\mathbf{X}e^{j\omega t}$ and $\mathbf{f} = \mathbf{F}e^{j\omega t}$. $\mathbf{X}$ and $\mathbf{F}$ are two amplitude vectors and contain one element for each degree of freedom (DOF). The elements of $\mathbf{X}$ are the displacement amplitudes of the respective DOF as a function of ω and the elements of $\mathbf{F}$ are the amplitudes of the force, again depending on ω, acting at location and in direction of the corresponding DOF. Since the harmonic part is available on both sides, we can ignore it and the equation of motion can be rewritten

$$\mathbf{X} = \left(\mathbf{K} - \omega^2\mathbf{M}\right)^{-1}\mathbf{F}, \tag{A.2}$$

where $\mathbf{X}$ and $\mathbf{F}$ means in practice $\mathbf{X}(\omega)$ and $\mathbf{F}(\omega)$, but are shorten for simplification, and ω is a diagonal matrix. Equation (A.2) is the direct frequency response analysis. The term $\mathbf{K} - \omega^2\mathbf{M}$ needs to be calculated for each frequency. To calculate the response to any excitation force $\mathbf{F}(\omega)$, we need to solve the eigenvalue problem:

$$\left(\mathbf{K} - \omega^2\mathbf{M}\right)\mathbf{X} = 0, \tag{A.3}$$

or

$$\left(\mathbf{M}^{-1}\mathbf{K}\right)\mathbf{X} = \omega^2\mathbf{X} = \lambda\mathbf{X}. \tag{A.4}$$

This equation says that each sounding object has a structure-related set of eigenvalues λ, which are simply connected to the system's frequencies. To extract the eigenvalues, the following condition has to be fulfilled

$$\det\left(\mathbf{K} - \omega^2\mathbf{M}\right) = 0. \tag{A.5}$$

Solving (A.5) implies finding the roots of a polynomial, which correspond to the eigenvalues $\boldsymbol{\lambda}$. The latter can then be replaced in (A.3):

$$\left(\mathbf{K} - \lambda\mathbf{M}\right)\Psi = 0, \tag{A.6}$$

Ψ is the matrix of eigenvectors, or eigenfunctions, where the column $\mathbf{r}$ is the vector related to the eigenvalue ω_r^2. The eigenvectors define the mode shapes linked to the corresponding frequency of the system.

If the frequencies are unique, many eigenvectors can be extracted for a given eigenvalue and all are proportional. Thus, the information enclosed in the eigenvectors is not the absolute amplitude but a ratio between the amplitudes in the degrees of freedom. For this reason, the eigenvectors are often normalized according to a reference. Due to the orthogonal property of the eigenvectors, $\Psi^T\Psi = \mathbf{I}$. Consequently, $\Psi^T\mathbf{M}\Psi$ and $\Psi^T\mathbf{K}\Psi$ are diagonal matrices, and are, respectively, called the modal mass and the modal stiffness of the system, because the ratio between modal stiffness and modal mass gives the matrix of eigenvalues. A very suitable reference choice is to scale the eigenvectors so that the modal mass matrix becomes an identity matrix. From (A.2), we can write:

$$\begin{aligned}\Psi^T\left(\mathbf{K} - \omega^2\mathbf{M}\right)\Psi &= \Psi^T\frac{\mathbf{F}(\omega)}{\mathbf{X}(\omega)}\Psi,\\ \left(\lambda - \omega^2\mathbf{I}\right) &= \Psi^T\frac{\mathbf{F}(\omega)}{\mathbf{X}(\omega)}\Psi,\end{aligned} \tag{A.7}$$

and finally

$$\mathbf{X}(\omega) = \Psi\left(\lambda - \omega^2\mathbf{I}\right)^{-1}\Psi^T\mathbf{F}(\omega). \tag{A.8}$$

Equation (A.8) simply expresses that the response $\mathbf{X}(\omega)$ can be calculated by surimposing a set of eigenmodes weighted by the excitation frequency, multiplied with an excitation load vector $\mathbf{F}(\omega)$.

Properties of Eigenvalues and Eigenvectors. The orthogonality of modes expresses that each mode contains information which the other modes do not have, and consequently a given mode cannot be built from the others. On the other hand, solutions of geometrically symmetric systems often give pairs of multiple eigenmodes.

Boundary conditions are settled simply by prescribing the value of certain degrees of freedom resolved in the displacement vector. As an example, a structure rigidly attached to the ground will show null DOFs around the support point. Consequently, the elements in the mode shapes corresponding to these DOFs will always be zero and will not need to be solved.

B. Damping

We now consider a damped system, and in particular the proportional damping model which assumes that the damping can be expressed proportional to the stiffness and mass matrix (Raleigh damping), that is, $\mathbf{C} = \alpha_1\mathbf{K} + \alpha_2\mathbf{M}$. In consequence, the eigenvalues of the proportional damped system are complex and can be expressed according to the eigenvalues of the undamped case

$$\lambda_r' = \omega_r^2\left(1 + j\eta_r\right), \tag{B.1}$$

where the imaginary part contains the loss factor η_r.

The modal superposition is thus given by

$$\mathbf{X}(\omega) = \Psi\left(\lambda - \omega^2\mathbf{I} + j\eta\lambda\right)^{-1}\Psi^T\mathbf{F}(\omega). \tag{B.2}$$

Equation (B.2) enables us to determine entire response velocity fields that cause the surrounding medium to vibrate and to generate sound.

Acknowledgments

This work was partly funded by Eden Games (http://www.eden-games.com/), an *ATARI* Game Studio in Lyon, France. C. Frisson is supported by numediart (http://www.numediart.org/), a long-term research program centered on digital media arts, funded by Région Wallonne, Belgium (Grant no. 716631). The authors would like to thank Nicolas Tsingos for his input on an early draft. Special thanks to Michaël Adam and Florent Falipou for their expertise in the *SOFA Framework*.

References

[1] J. F. O'Brien, C. Shen, and C. M. Gatchalian, "Synthesizing sounds from rigid-body simulations," in *Proceedings of ASM SIGGRAPH Symposium on Computer Animation*, pp. 175–181, July 2002.

[2] J.-M. Adrien, "The missing link: modal synthesis," in *Representations of Musical Signals*, pp. 269–298, MIT Press, Cambridge, Mass, USA, 1991.

[3] C. Cadoz, A. Luciani, and J. Florens, "Responsive input devices and sound synthesis by simulation of instrumental mechanisms: the CORDIS system," *Computer Music Journal*, vol. 8, no. 3, pp. 60–73, 1984.

[4] F. Iovino, R. Caussé, and R. Dudas, "Recent work around modalys and modal synthesis," in *Proceedings of the International Computer Music Conference (ICMC '97)*, pp. 356–359, 1997.

[5] P. R. Cook, *Real Sound Synthesis for Interactive Applications*, A. K. Peters, 2002.

[6] K. Van den Doel, P. G. Kry, and D. K. Pai, "Foley automatic: physically-based sound effects for interactive simulation and animation," in *Proceedings of the Computer Graphics Annual Conference (SIGGRAPH '01)*, pp. 537–544, August 2001.

[7] N. Raghuvanshi and M. C. Lin, "Interactive sound synthesis for large scale environments," in *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3d '06)*, pp. 101–108, March 2006.

[8] C. Bruyns-Maxwell and D. Bindel, "Modal parameter tracking for shape-changing geometric objects," in *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx '07)*, 2007.

[9] Q. Zhang, L. Ye, and Z. Pan, "Physically-based sound synthesis on GPUs," in *Proceedings of the 4th International Conference on Entertainment Computing*, vol. 3711 of *Lecture Notes in Computer Science*, pp. 328–333, 2005.

[10] N. Bonneel, G. Drettakis, N. Tsingos, I. Viaud-Delrnon, and D. James, "Fast modal sounds with scalable frequency-domain synthesis," in *Proceeding of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '08)*, August 2008.

[11] D. Menzies, "Physical audio for virtual environments, phya in review," in *Proceedings of the International Conference on Auditory Display (ICAD)*, G. P. Scavone, Ed., pp. 197–202, Schulich School of Music, McGill University, 2007.

[12] J. F. O'Brien, P. R. Cook, and G. Essl, "Synthesizing sounds from physically based motion," in *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pp. 529–536, Los Angeles, Calif, USA, August 2001.

[13] R. J. McAulay and T. F. Quatieri, "Speech analysis/ synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[14] K.-J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1982.

[15] D. L. James, J. Barbič, and D. K. Pai, "Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 987–995, 2006.

[16] J. N. Chadwick, S. S. An, and D. L. James, "Harmonic shells: a practical nonlinear sound model for near-rigid thin shells," in *Proceedings of ACM Transactions on Graphics*, 2009.

[17] M. Nesme, Y. Payan, and F. Faure, "Animating shapes at arbitrary resolution with non-uniform stiffness," in *Proceedings of Eurographics Workshop in Virtual Reality Interaction and Physical Simulation (VRIPHYS '06)*, 2006.

[18] J. R. Shewchuk, "A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations," in *Proceedings of the 14th Annual Symposium on Computational Geometry*, pp. 76–85, June 1998.

[19] M. Nesme, P. G. Kry, L. Jeřábková, and F. Faure, "Preserving topology and elasticity for embedded deformable models," *ACM Transactions on Graphics*, vol. 28, no. 3, article no. 52, 2009.

[20] K. van den Doel and D. K. Pai, "Modal Synthesis for Vibrating Objects," in *Audio Anecdotes III: Tools, Tips, and Techniques for Digital Audio*, pp. 99–120, A. K. Peter, 3rd edition, 2006.

[21] C. Bruyns, "Modal synthesis for arbitrarily shaped objects," *Computer Music Journal*, vol. 30, no. 3, pp. 22–37, 2006.

[22] D. Menzies, "Phya and vfoley, physically motivated audio for virtual environments," in *Proceedings of the 35th AES International Conference*, 2009.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 617325, 13 pages
doi:10.1155/2010/617325

Research Article

Distortion Analysis Toolkit—A Software Tool for Easy Analysis of Nonlinear Audio Systems

Jyri Pakarinen

Department of Signal Processing and Acoustics, School of Science and Technology, Aalto University, P.O. Box 13000, 00076 Aalto, Finland

Correspondence should be addressed to Jyri Pakarinen, jyri.pakarinen@tkk.fi

Received 27 February 2010; Revised 30 April 2010; Accepted 17 May 2010

Academic Editor: Udo Zoelzer

Several audio effects devices deliberately add nonlinear distortion to the processed signal in order to create a desired sound. When creating virtual analog models of nonlinearly distorting devices, it would be very useful to carefully analyze the type of distortion, so that the model could be made as realistic as possible. While traditional system analysis tools such as the frequency response give detailed information on the operation of linear and time-invariant systems, they are less useful for analyzing nonlinear devices. Furthermore, although there do exist separate algorithms for nonlinear distortion analysis, there is currently no unified, easy-to-use tool for rapid analysis of distorting audio systems. This paper offers a remedy by introducing a new software tool for easy analysis of distorting effects. A comparison between a well-known guitar tube amplifier and two commercial software simulations is presented as a case study. This freely available software is written in Matlab language, but the analysis tool can also run as a standalone program, so the user does not need to have Matlab installed in order to perform the analysis.

1. Introduction

Since the 1990s, there has been a strong trend in the digital audio effects community towards virtual analog modeling, that is, mimicking the sound of old analog audio devices using digital signal processing (DSP). Guitar tube amplifier emulation [1] has been an especially vibrant area of research with several commercial products.

There are roughly two methodological approaches for designing virtual analog models: the black-box- and the white-box methods. In the former, the designer treats the original audio device (called target in the following) as an unknown system and tries to design the model so that it mimics only the input/output relationship of the target, without trying to simulate its internal state. In the white-box approach, the designer first studies the operation logic of the device (often from the circuit schematics) and tries to design the model so that it also simulates the internal operation of the target. While both approaches can yield satisfactory results in many cases, there is still room for improvement in the sound quality and model adjustability, if the virtual analog models are to act as substitutes for the original analog devices.

In all virtual analog modeling, the target and model responses need to be carefully analyzed. This is obviously essential in the black-box approach, but it is also an unavoidable step in white-box modeling, since even circuit simulation systems need to have their component values adjusted so that the model response imitates the target response. While some part of this analysis can be conducted simply by listening to the models, there is a clear need for objective analysis methods. If the target system can be considered linear and time invariant (LTI), it is relatively straightforward to analyze the magnitude and phase responses of the target and model and try match them as closely as possible. For distorting systems, however, linear magnitude and phase responses give only partial information on how the system treats audio signals, since distortion components are not analyzed.

Analysis of mildly distorting audio systems, such as loudspeakers, have traditionally consisted of simple total harmonic distortion (THD) measurements, where a static

sine signal is fed to the system as an input, and the harmonic distortion component levels are summed and normalized to the amplitude of the fundamental component. Even though THD gives a vague indication on the general amount of static harmonic distortion, it tells nothing on the type of this distortion. Thus, THD is a poor estimate for analyzing distorting audio effects. Although more informative analysis techniques have been designed by the scientific community, there is currently no unified, easy-to-use tool for simultaneously conducting several distortion analysis measurements.

2. Distortion Analysis Toolkit

2.1. Overview. The distortion analysis toolkit (DATK) is a software tool for analyzing the distortion behavior of real and virtual audio effects. It operates by first creating a user-defined excitation signal and then analyzes the responses from the devices. The DATK package includes five analysis techniques, which can be further augmented with additional user-defined analysis functions. The DATK is developed in Matlab language, but standalone operation is also possible. Since a distortion measurement on an audio effects device usually consists of making several individual recordings while varying some control parameter and keeping the other parameters fixed, the DATK is designed for batch processing. This means that the DATK creates an excitation signal as an audio file according to the specifications set by the user. Next, the user plays this excitation to the device under test (DUT) and records the output on any playback/recording software. The DUT response to its control parameter variations can be tested by changing the control parameter values and rerecording the response. As a result, the user is left with a set of response files that each have different control parameter settings.

The analysis part is carried out by pointing the excitation file and the set of response files to the DATK, after which it plots analysis figures, so that each analysis type is shown as a separate figure, and the analysis results from each response file are illustrated with subfigures. Thus, it is easy to interpret the effect of control parameter variations on the system response, since each subfigure shows the response to a different value of the varied parameter. The analysis figures can also be saved as.fig files even in standalone mode, allowing further editing with Matlab. The DATK runs on any modern computer, and only requires the use of a simple playback/recording software for playing the excitation signal and recording the responses. For measuring software plugins, the playback/recording software should be able to act as a plugin host. For measuring physical effects devices, an audio interface with adjustable playback/recording gains should be used. Sections 2.2 through 2.6 discuss the analysis functions currently included in the DATK, while Section 2.7 discusses how to develop additional analysis functions. Since the user defines the desired analysis techniques and their parameters by writing a text file, where the different analysis types (or same analysis types with different parameters) are written as separate lines, Sections 2.2 through 2.6 also introduce the syntax for performing each analysis type. The actual use of the DATK software is illustrated in Section 3 with a case study.

2.2. Sine Analysis. One of the simplest distortion analysis techniques is to insert a single sinusoid signal with fixed amplitude and frequency into the system, and plot the resulting spectrum. The advantage of this analysis type is that it is easy to understand and the results are straightforward to interpret. The disadvantage is that this analysis gives only information on how the system treats a single, static sine signal with given amplitude and frequency. In particular, it tells nothing about the dynamic behavior of the system. The DATK sine analysis function is invoked using the syntax

```
SineAnalysis(frequency,duration)
```

where the `frequency` denotes the frequency of the analysis sine in Hertz, and `duration` is the signal duration in seconds. The amplitude of the sine is set to unity. In the analysis phase, the DATK plots the resulting system response spectrum in the audio frequency range so that the amplitude of the fundamental frequency component (i.e., the component at the same frequency as the analysis sine) is 0 dB. The fundamental component is further denoted with a small circle around its amplitude peak, which helps in locating it if the spectrum has high-amplitude distortion components at subharmonic- or otherwise low frequencies.

2.3. Logsweep Analysis. The logsweep analysis technique is an ingenious device for analyzing static harmonic distortion as a function of frequency. Introduced by Farina in 2000 [2], the basic idea behind the logsweep analysis technique is to insert a logarithmic sine sweep signal with fixed amplitude to the system, and then convolve the time-reversed and amplitude weighted excitation signal with the system response. As a result, one obtains an impulse response signal, where the response of each harmonic distortion component is separated in time from the linear response and each other. If the duration of the logsweep signal is long enough, the time separation between the harmonic impulse responses is clear and it is straightforward to cut the individual harmonic responses and represent them in the frequency domain. As a result, one obtains magnitude response plots for the linear response and each harmonic distortion component.

The DATK logsweep analysis function uses the syntax

```
LogsweepAnalysis(start_freq,end_freq,
duration,number_of_harmonics)
```

where `start_freq` and `end_freq` are the frequency limits (in Hz) for the logarithmic sweep, and `duration` is the sweep duration in seconds. The parameter `number_of_harmonics` sets the number of harmonic components to plot in the analysis phase, so that value 3, for example, would correspond to the linear response and the 2nd and 3rd harmonic distortion components to be drawn. As a rule of thumb, the signal to noise ratio (SNR) decreases with increasing harmonic numbers, so it is often not advisable to try to analyze a large number of harmonics

(e.g., >6), unless long excitation signals (several seconds) are used.

In the analysis phase, the DATK plots the linear and harmonic distortion component magnitude responses as a function of frequency (in the frequency range from `start_freq to end_freq`) in a single figure. The response curves are illustrated with different colors and each curve has a number denoting the harmonic order, starting from 1 for the linear response, 2 for the 2nd harmonic, and so forth. The magnitude offset of the responses is set so that the magnitude of the linear response is 0 dB at low frequencies. The magnitude responses of the harmonic distortion components are shifted down in frequency by the order of their harmonic number. This means that looking at the analysis figure at a given frequency, say at 1 kHz, curve 1 gives the magnitude of the linear component, curve 2 gives the magnitude of the 2nd harmonic (which actually has the frequency of 2 kHz), curve 3 gives the magnitude of the 3rd harmonic (residing at 3 kHz), and so forth. This will be further illustrated in Section 3.6. The harmonic impulse responses are windowed using a Blackman window prior to moving them into the frequency domain, for obtaining smoother magnitude response plots [3].

The advantage of this measurement technique over the sine analysis (Section 2.2) is that it displays more information in a single figure by drawing the magnitude responses as a function of the (fundamental) frequency on the full audio frequency range. For analyzing the magnitude of high-order distortion components, however, the sine analysis is usually better since it displays the entire audible spectrum in one figure, while the logsweep analysis can successfully be used to track only the low-order distortion components, due to clarity reasons and the aforementioned SNR issue. Another drawback of the logsweep analysis technique is that long excitation signals can take a long time to analyze due to the relatively slow computation of long convolutions.

2.4. Intermodulation Distortion Analysis. When a multitone signal is clipped, intermodulation distortion (IMD) occurs. This means that the distortion creates signal components not only to the integer multiples of the original tones, but also to frequencies which are the sums and differences of the original signal components and their harmonics. In musical context, IMD is often an undesired phenomenon, since IMD components generally fall at frequencies which are not in any simple harmonic relation to the original tones, making the resulting sound noisy and inharmonic.

Traditionally, the IMD of distorting systems has been measured by feeding a sum of two sinusoids with different frequencies into the system and plotting the resulting spectrum. Although this type of analysis gives an exact indication of the frequencies and amplitudes of the distortion components for a given pair of input signal frequencies and amplitudes, it is not very useful in determining the overall level of the IMD. It should be noted that with distorting audio effects, it is often useful to know the exact frequency and amplitude behavior of the harmonic distortion components, since there are some strong opinions regarding the harmonic content and the resulting sound, such as "the vacuum tube sound consists mainly of even harmonics", or that "the 7th harmonic is something to be avoided". With IMD, however, it is probably more useful to have a single number indicating the overall amount of distortion, since IMD is generally considered an undesired phenomenon, with no specific preferences on any particular IMD components. Furthermore, traditional IMD measurements ignore the dynamic behavior of the intermodulation mechanism, so that dynamic or transient intermodulation (DIM/TIM) [4] distortion components are left unnoticed.

The IMD analysis of the DATK software is based on the measurement technique introduced by Leinonen et al. [5], which measures both the DIM distortion and static IMD. In this measurement approach, a square wave signal with a fundamental frequency f_1 is summed with a lower-amplitude sinusoid having a frequency f_2, so that $f_1 < f_2$. Next, this signal is inserted to the DUT, and the amplitude of the IMD components at audio frequency range is compared to the amplitude of the sinusoid. As a result, a percentage ratio between the root-mean-square (RMS) IMD components and the low-amplitude sinusoid riding on top of the square wave is obtained. Thus, an IMD percentage value of 100 % would mean that the RMS IMD is equal in magnitude to the frequency component that causes the IMD in the first place. The static IMD is measured in a very similar way, the only difference being that instead of a square wave signal, a triangular waveform is used.

For measuring the IMD, the DATK creates two signals: one containing the square wave and the sine wave (for measuring DIM), and one containing the triangle wave and the sine (for measuring static IM). In order to estimate the IMD as a function of input signal amplitude, both of these signals are weighted by a linear amplitude ramp, so that the overall signal amplitude increases with time, although the amplitude ratios between the square (or triangle) wave and the sine wave remain unchanged. The DATK IMD analysis is called with the function

```
IMDAnalysis(sine_freq,sq_freq,sine_ampl,
sq_ampl,duration)
```

where `sine_freq` and `sine_ampl` are the frequency (Hz) and amplitude of the sinusoid, respectively, and `sq_freq` and `sq_ampl` are the fundamental frequency and amplitude of the square wave. The triangle wave uses the same frequency and amplitude parameters as the square wave. The frequencies of the sine and square waves should be selected so that they are not in an integer relation with each other or the sampling frequency of the system.

The final parameter, `duration`, sets the length (in seconds) of each of the IMD test signals. The amplitude parameters are normalized so that the ratio between the sine and the square wave amplitudes remains `sine_ampl/sq_ampl`, but the overall signal amplitude reaches unity at the end of the test signal ramp. For avoiding digital aliasing of the square and triangle wave signals, the DATK creates them using the differentiated parabolic waveform technique, introduced in [6]. At the analysis phase, the DATK analyzes the system's

output spectra for the IMD test signals and evaluates the DIM and IM percentages.

The advantage of the IMD measurement suggested in [5] over the traditional IMD measurements (i.e. simply plotting the output spectra), is that it provides two intuitive parameters concerning the IMD level: the IM percentage, which tells the amount of IMD associated with relatively static signals (such as the sound of a violin), and the DIM percentage, which tells the amount of IMD associated with transient signals (such as the sound of percussions).

2.5. Transient Response Analysis. Measuring the transient behavior of distorting audio devices is difficult in general. Typically, the transient response depends heavily on the amplitude and frequency content of the test signal, so it is usually not possible to generalize the results for other types of input signals. Nevertheless, it is still useful to compare the input/output waveforms of a distorting device when a transient signal is used as input, for example because certain types of dynamic distortion (such as the blocking distortion, as in [7]) can be identified from the waveforms.

The DATK transient response analysis operates by creating a sine burst signal, where the first cycle (called transient) has a larger amplitude than rest of the sine burst (called tail). The DATK transient response analysis function uses the syntax

```
TransientAnalysis(tail_ampl,freq,
duration,cycles_to_draw)
```

where `tail_ampl` denotes the amplitude of the tail of the sine burst following the transient, and should be a positive number smaller than one (the amplitude of the transient is always unity). Parameters `freq` and `duration` set the frequency (Hz) and duration (sec) of the sine burst, respectively. The parameter `cycles_to_draw` sets the initial zoom limit in the analysis figure, so that dynamic distortion effects are easy to see. For example, a value of 10 would result in a waveform analysis figure, where 10 cycles of the response wave were displayed. The optimal value for the `cycles_to_draw` parameter obviously depends on the dynamic distortion characteristics of the device. If the DUT acts only as a static nonlinearity, the amplitude envelope, horizontal offset, and shape of the tail should remain constant in the analysis figure.

2.6. Aliasing Analysis. The DATK also has a tool for estimating the effect of signal aliasing in digital distorting devices. Nonlinear distortion always expands the signal spectrum by creating harmonic (and other) distortion components. This is problematic in a digital implementation, since the distortion components exceeding the Nyquist frequency (half the sampling rate) will alias back to the baseband, possibly resulting in an inharmonic and noisy signal. There are several techniques for avoiding aliasing within digital distortion [1], oversampling being probably the most popular.

The aliasing analysis tool in the DATK operates by creating a high-frequency sine signal, and analyzing the spectrum of the system response. This analysis technique, introduced

Figure 1: The custom-built AC30 tube guitar amplifier (photo by Ari Viitala).

in a recent journal article [7], tracks the nonaliased harmonic distortion components and fits a simple auditory spectrum curve on them, for estimating the frequency masking effect of the baseband signal. More specifically, the auditory spectrum curve is estimated by fitting a gammatone filterbank [8] magnitude response curve on top of the nonaliased signal so that the center frequencies of the filters are aligned with the frequencies of the baseband components. A fixed 10 dB offset between the peak of each sinusoidal signal component and the corresponding gammatone filter is applied since it was empirically found to match well with many distorted signals. Next, the effect of the hearing threshold is added by estimating the F-weighting function [9] for audio frequencies, together with a user-defined sound pressure level (SPL) value. The net effect of the masking curves and hearing threshold is obtained by taking the maximum value of the filter magnitude responses and the F-weighting function for each frequency.

Finally, the DATK creates a residual spectrum by removing the original sine and the nonaliased distortion components, and displays the residual spectrum in the same figure with the auditory spectrum estimate. Since the residual spectrum consists of the aliased signal components (and noise), one can assume that the aliasing artifacts (or noise) are audible, if the residual spectrum exceeds the auditory spectrum estimate in magnitude at any frequency. It should be noted that although there exist more sophisticated techniques (e.g., [10, 11]) for estimating the auditory spectrum of complex tones, the use of a relatively simple auditory model in this context is enough to characterize the perceptual point of view. Evaluation in detail requires a detailed model compared against a careful formal listening test, which falls beyond the intended scope of the DATK tool.

The DATK aliasing analysis function is invoked by calling

```
AliasingAnalysis(freq,duration,SPL)
```

where `freq` is the frequency of the sine. The sine frequency should be selected from the normal frequency range defined for the system under measurement. For example, measuring the aliasing of a digital guitar effects device at a frequency of 16 kHz would not yield realistic results, since the electric guitar signal usually has very little energy above 15 kHz [12]. The duration parameter sets the `duration` of the sine signal

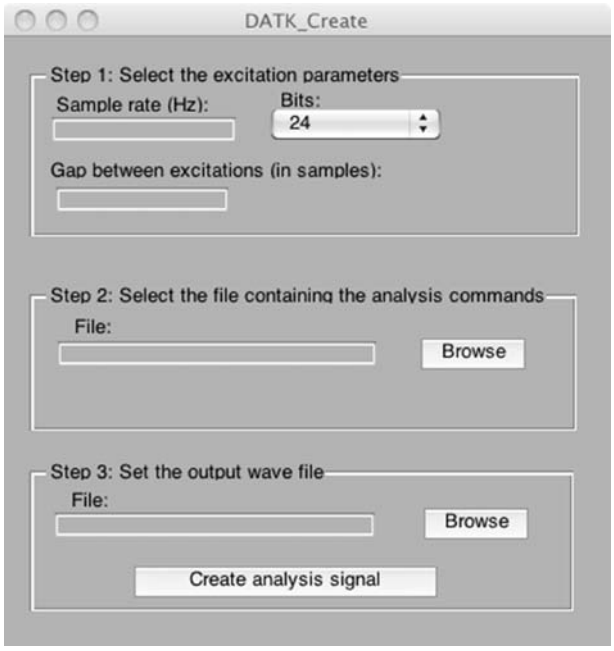


Figure 2: Dialog box for creating the analysis signal.

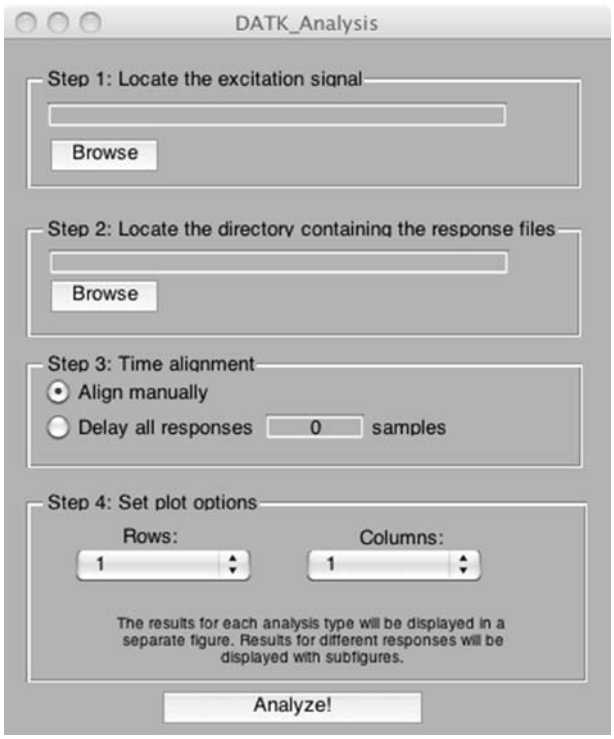


Figure 3: A dialog box for analyzing the system responses.

(in seconds), and the SPL parameter is the estimated F-weighted sound pressure level at which the distorted sine signal would be played.

2.7. Developing Additional Analysis Functions. The user can append the DATK's collection of analysis tools by developing his or her own signal generation and analysis functions. Although the DATK can be operated as a standalone program, developing additional analysis tools requires the use of Matlab. Appending the DATK to include custom user-defined analysis tools is relatively simple, provided that the user is familiar with basic signal processing tasks on Matlab. The user-defined analysis tools should comply with some consistency rules, namely that

(i) there is a function for creating an excitation signal, possibly according to parameters set by the run-time user,

(ii) there is an analysis function which accepts a response signal and plots an analysis figure accordingly,

(iii) any additional parameters that the analysis function requires are generated as metadata by the excitation creation function and stored in a file.

More detailed instructions on developing custom analysis tools for DATK can be found on the DATK website: http://www.acoustics.hut.fi/publications/papers/DATK/. After developing the custom analysis tools, the user can easily compile a standalone version of the updated DATK using Matlab's `deploytool`-function.

3. Case Study: Measurement on the Real and Virtual AC30 Guitar Amplifiers

This section illustrates the use of the standalone DATK software by measuring the distortion characteristics of an AC30 guitar tube amplifier and two of its virtual analog versions. The main purpose of this section is to provide an example case for producing distortion measurements using the DATK, rather than to perform a detailed interpretation of the operation of the devices. Thus, the emphasis is on explaining the measurement procedure, and what is the useful information in the analysis results, without trying to explain why the amplifier or its virtual versions have a certain effect on the signal.

3.1. VOX AC30. The AC30 is an iconic guitar tube amplifier, first introduced by the VOX company in 1959 [13]. This class AB [14] amplifier uses four cathode-biased EL-84 tubes in the output stage. In 1961, a "Top Boost"-unit, an additional circuit box including treble and bass controls and an extra gain stage was introduced for the AC30. Due to its immediate popularity among users, the Top Boost unit was integrated into the AC30 circuitry from 1963 onwards. The distinct "jangly" sound of the AC30 amplifier can be heard on many records from several bands such as the Beatles, The Rolling Stones, The Who, Queen, R. E. M., and U2, from the last five decades.

A custom-built AC30 amplifier, illustrated in Figure 1, was chosen as a test example for the DATK. This amplifier (referred to as "real AC30" in the following) is a relatively faithful reproduction from the original AC30 circuit schematics, although the tremolo circuit has been omitted. In addition to the real AC30, the distortion measurements

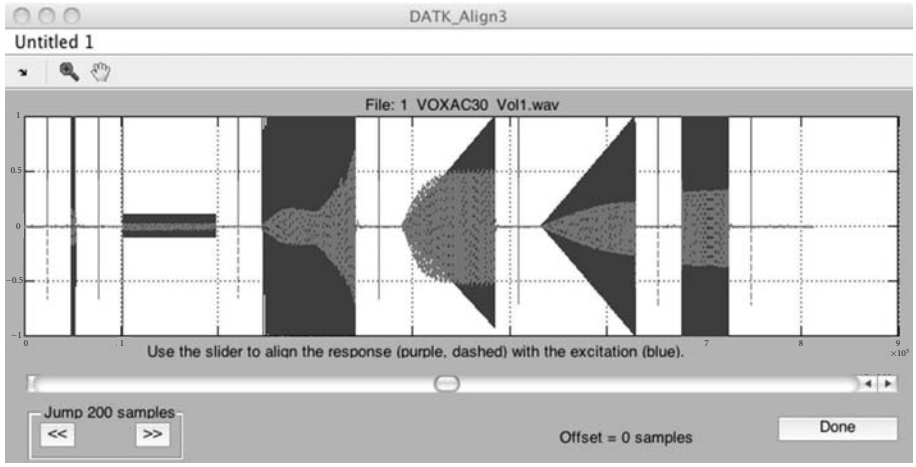


Figure 4: Graphical user interface for manual alignment of the analysis signal.

were conducted on two commercial software emulators, namely the Top30 unit of the Overloud TH1 [15] plugin, and the FOX ACS-45 unit of the Peavey ReValver [16] plugin (abbreviated TH1 AC30 and ReValver AC30, resp., in the following).

3.2. Measurement Setup. The distortion behavior of the real AC30 was measured with the DATK software running on a 2.4 GHz MacBook (Intel Core 2 Duo, 4 GB RAM) laptop computer with M-Audio Firewire 410 as the audio interface. The Reaper software [17] was used in playing the excitation and recording the responses. An 8 Ω power resistor was used as a load for the amplifier. The amplifier output signal was obtained by recording the voltage over the load resistor using a custom-built attenuator/buffer circuit to eliminate the impedance coupling between the amplifier and the audio interface. It should be noted, that although real tube amplifiers usually require a separate attenuator/buffer circuit, many other distortion devices, such as effects pedals, may directly be connected to the audio interface. The high-impedance bright channel was used on the amplifier, and the tone knobs were all set to 12 o'clock. Different responses were measured by varying the "Volume bright" knob, which in practice controls the gain for the bright channel. On the software plugins also, the bright channel was selected (called "Brilliant" on the ReValver AC30), all tone controls were set to 12 o'clock, and different recordings were made while varying the channel gain. It should be noted that since the real AC30 was operated without a loudspeaker, the loudspeaker emulation was switched off for both virtual plugins.

3.3. Creating the Analysis Signal. A complete analysis was performed on the three systems. The five functions were stored in a text file, with the following parameter values:

```
SineAnalysis(1000,0.1),
TransientAnalysis(1/10,1000,2,20),
LogsweepAnalysis(20,20000,2,5),
IMDAnalysis(6000,1270,1,4,2),
AliasingAnalysis(5000,1,80),
```

This file was read during the analysis by DATK. Next, the excitation signal was created by double-clicking the `DATK_Create` program icon, which opens a dialog box illustrated in Figure 2. At step 1, a sampling frequency of 48000 Hz and a bit length equal to 24 were selected from the dialog box. A gap of 24000 samples (corresponding to half a second with the selected sample rate) was chosen between the individual excitations. The purpose of this segment of silence between the measurement signal bursts is to ensure that the different measurements do not affect each other, that is, that the previous response signal has faded before a new one begins. At step 2, the recently created text file was selected using the "Browse"-button, and at step 3, the name and path for the excitation file was selected. Finally, the "Create analysis signal"-button was pressed, resulting in a command-line message denoting that the excitation signal was successfully created.

3.4. Recording. The excitation signal was imported by the recording software, and it was played to the real AC30 while recording its output onto another track. Three recordings were made with different settings for the "volume bright" knob: 9 o'clock (low-gain), 12 o'clock (middle gain), and full (high gain). The three outputs were individually saved as.wav files within the same directory. Next, the TH1 plugin with only the Top30 unit enabled (and the channel gain knob set to 9 o'clock) was added as an effect on the excitation track, and the response was saved as a wave file to the same directory as the real AC30 response files. The TH1 AC30 response recordings were repeated two times with the virtual channel gain knob set to 12 o'clock and full, in order to obtain response wave files with all three gain settings. Finally, the TH1 plugin was replaced with the Peavey ReValver plugin with only the FOX ACS-45 unit enabled, and the same operations (adjust gain, render output wave file, repeat) were carried out three times. Thus in the end,

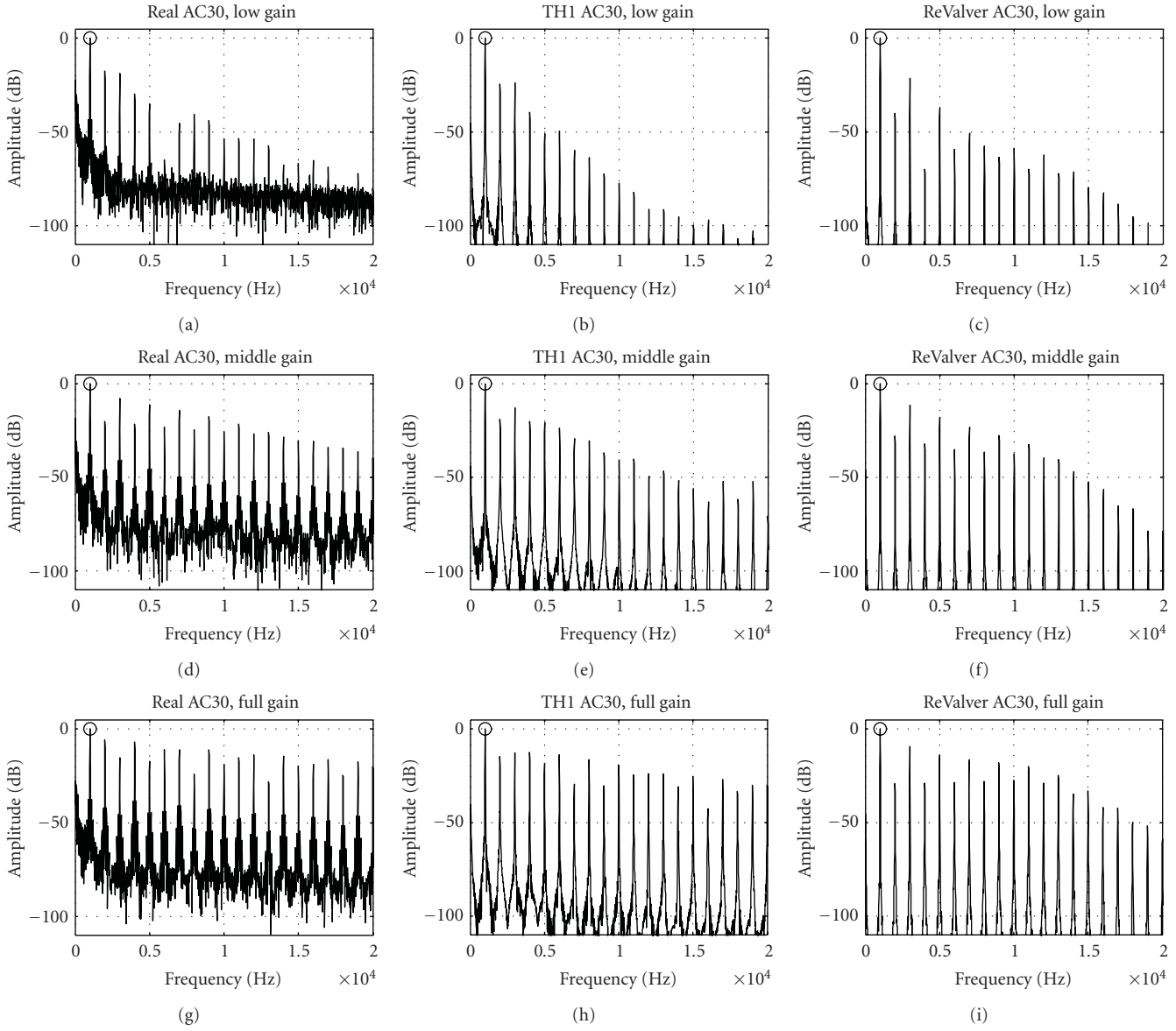


Figure 5: Response spectra to a 1 kHz sinusoid for the real AC30 (left column), the Top30 unit of the TH1 software plugin (middle column), and the FOX ACS-45 unit of the ReValver software plugin (right column). The user adjustable gain knob was first set to 9 o'clock (top row), then to 12 o'clock (middle row), and finally to full (bottom row). As expected, the real AC30 has a higher noise floor than the digital versions.

the response directory contained 9 response signals as.wav files with sample rate of 48 kHz and 24-bit resolution.

3.5. Signal Analysis. The analysis part of the DATK was started by double-clicking the `DATK_Analyze`-program icon, which opens a dialog box illustrated in Figure 3. First, the wave file containing the excitation signal and the directory containing the response files were located using the "Browse" buttons at steps 1 and 2. The next pane, step 3, defines the type of time alignment between the excitation and response signals. The "Align manually" option, which allows the user to individually set the time alignment, was used. The other option allows the user to define a common delay (positive or negative) for all response files. A common delay value may be useful, for example if all responses are obtained from the same device since the signal delay is typically identical between different responses. Since there were three versions of the AC30 with three gain settings each, three rows and three columns were selected at step 4 to display the nine analysis subfigures. Finally, the distortion analysis was started by pressing the "Analyze!" button.

Since manual alignment was chosen at step 3 of the analysis dialog box, the DATK next opens a time alignment window, displayed in Figure 4. Here, the user's task is to move the response signal (denoted with a dashed purple line) in time using a slider, so that it coincides with the excitation signal (denoted with a solid blue line). The resolution of the time alignment is one sample, and the zoom option can be used in adjusting the time offset as carefully as possible. After

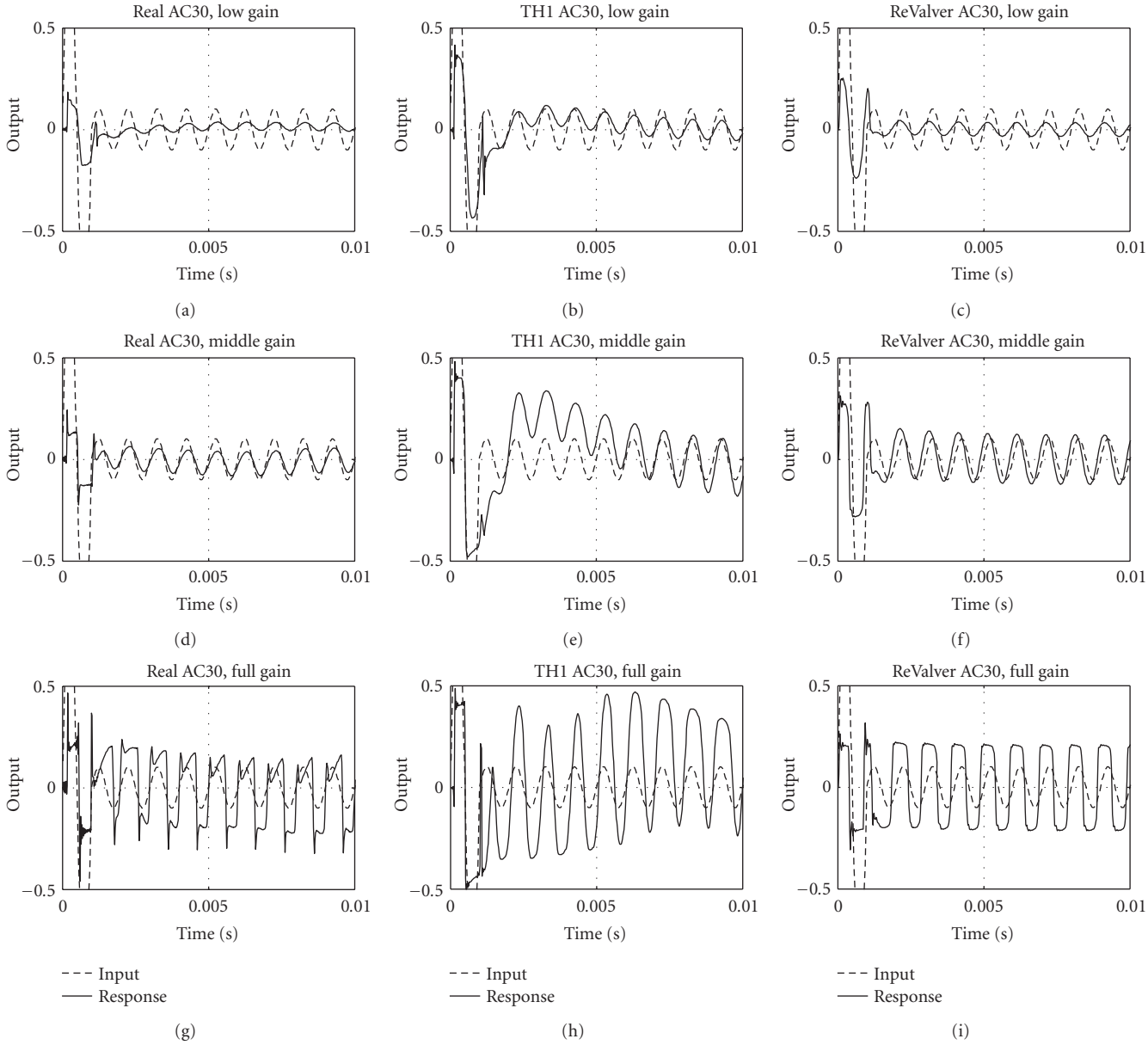


Figure 6: Transient analysis waveforms for the real AC30 (left column), the TH1 AC30 (middle column), and the ReValver AC30 (right column). The input signal is illustrated with a dashed line, while the solid line denotes the measured response. The gain knob was first set to 9 o'clock (top row), then to 12 o'clock (middle row), and finally to full (bottom row).

the user has aligned all the responses with the excitation, the DATK begins the analysis calculation and draws the result figures in the order defined in the analysis command text file.

3.6. Results. Prior to performing the analysis using DATK, the three AC30 variants were tested with an electric guitar. The output signals from all test cases were fed to headphones through a TH1 loudspeaker cabinet simulator in order to obtain a more natural electric guitar sound. This informal and highly subjective playing test revealed that the real AC30 had a certain "smooth" distortion type that was not entirely captured by either software plugins, although the TH1 AC30 managed to get closer to it than the ReValver AC30, which sounded somehow too "harsh", especially when using full gain.

The first analysis plot drawn by the DATK is the sine analysis figure, illustrated in Figure 5. With low-gain settings (top row), all three AC30 variants share a similar lowpass-type spectral envelope. The amplitude ratios of the first few harmonic components on the TH1 AC30 show a relatively good match with the real AC30, while the low-order even harmonics are more suppressed with the ReValver AC30. Interestingly, the 6th harmonic is heavily attenuated in the real AC30 response, while the virtual versions do not mimic this.

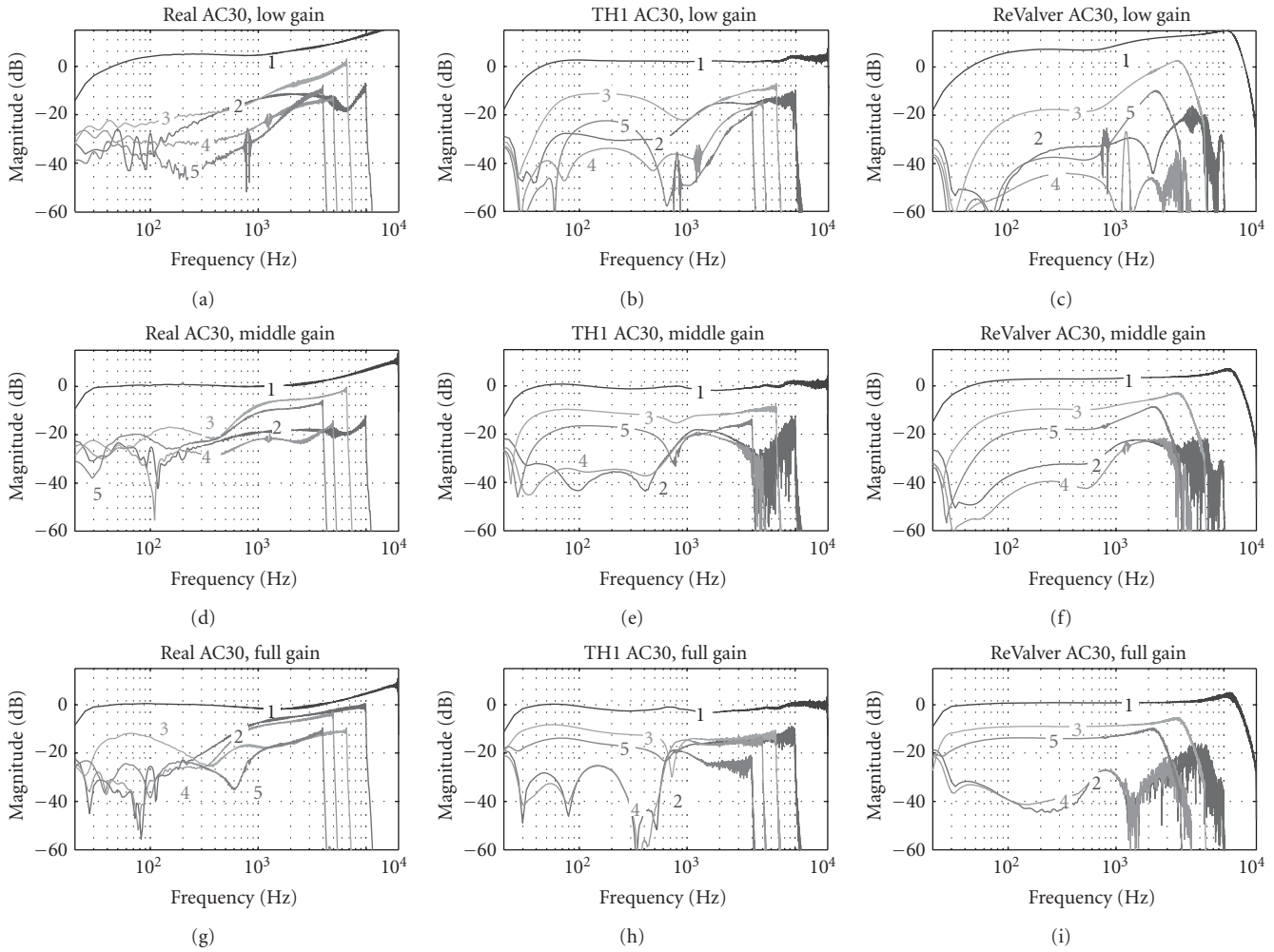


Figure 7: Logsweep analysis of the real AC30 (left column), the TH1 AC30 (middle column), and the ReValver AC30 (right column). The gain knob was first set to 9 o'clock (top row), then to 12 o'clock (middle row), and finally to full (bottom row). The numbered curves represent the order of the distortion, so that curve 1 corresponds to the linear magnitude response, curve 2 to the 2nd order harmonic distortion, curve 3 to the 3rd order harmonic distortion, and so forth. The 6th order and higher distortion components are not illustrated.

Increasing the gain level to medium (middle row), the even harmonic components become more suppressed in the real AC30 response, while the overall spectral envelope gets brighter. The ReValver AC30 response seems to simulate this even-order distortion component attenuation well. Both virtual plugins show a slightly stronger lowpass-effect than the real AC30 with medium gain. For full gain settings, the even-order distortion components exceed the odd-order components in the real AC30 response. For the ReValver AC30, the opposite happens: the odd-order components are significantly louder. As expected, the overall spectral envelope gets even brighter with increased gain in all three AC30 variants. Looking at all responses in Figure 5, it seems that the spectrum of the real AC30 response varies strongly on the gain level and this change is more subtle with the software versions.

The second analysis type produces the transient response plot, illustrated in Figure 6, where the dashed line denotes the input signal, while the response waveform is drawn with a solid line. The vertical axis has been zoomed in between −0.5–0.5 to better illustrate the dynamic effects. It can be seen in the figure that the initial transient is hard-clipped in nearly all cases. For low-gain settings (top row), the real AC30 shows a slight bias shift after the transient, without a strong effect on the waveform. The TH1 AC30 response experiences a slightly stronger bias shift, with a heavier distortion immediately after the transient, while the ReValver AC30 response is more static. For medium and full gain settings (middle and bottom row), the real AC30 shows a slight attenuation right after the transient, which might indicate blocking distortion [7]. The TH1 AC30 response shows an exaggerated bias shift due to the transient, while the ReValver AC30 response is again more static. With full gain, the real AC30 waveform is more severely distorted than with the software plugins.

The third analysis figure is the logsweep response plot, illustrated in Figure 7. Here, it can be seen that the linear response (curve 1) of the ReValver AC30 shows a similar

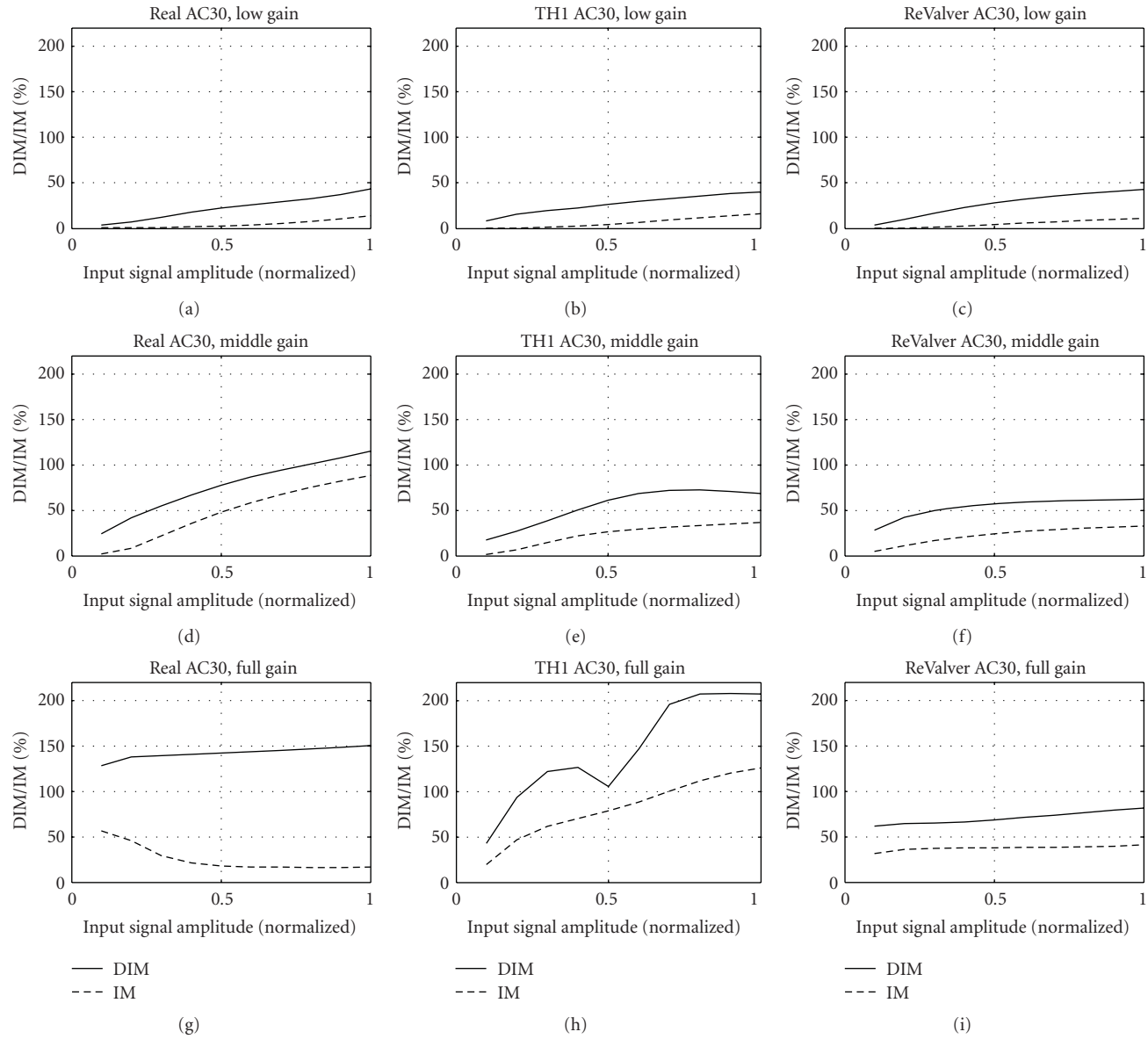


Figure 8: Intermodulation distortion analysis of the real AC30 (left column), the TH1 AC30 (middle column), and the ReValver AC30 (right column). The gain knob was first set to 9 o'clock (top row), then to 12 o'clock (middle row), and finally to full (bottom row). The solid line illustrates the amount of dynamic intermodulation distortion, while the dashed line denotes the amount of static intermodulation distortion, both as a function of the normalized input signal level.

highpass behavior in the audio range as the real AC30, while the TH1 AC30 response is flatter. For low and medium-gain settings, the distortion components on the real AC30 have a highpass nature. The even harmonics on the TH1 AC30 mimic this somewhat, while the odd harmonics are more pronounced in the low frequencies than with the real AC30. The odd harmonics on the ReValver AC30 have a highpass trend for low and medium-gain, but the even harmonics are attenuated more than in the real AC30. For full gain, the second and third harmonics on the real AC30 are very strong at high frequencies. Neither software plugins have such a high distortion in the kilohertz range as the real AC30. Furthermore, neither plugins mimic the boosting of the third harmonic and simultaneous suppression of the fifth harmonic at low frequencies (near 70 Hz).

The fourth analysis result created by the DATK is the intermodulation distortion analysis plot, illustrated in Figure 8. It shows the dynamic intermodulation (DIM) distortion percentage with the solid line, and the static intermodulation (IM) distortion percentage with a dashed line, both as a function of the normalized input signal amplitude. As can be seen in the figure, the DIM distortion exceeds the static IM distortion in all cases. For low- and medium gain settings (top and middle row), the DIM/IM distortion increases monotonically with input signal amplitude in nearly the same way for all three AC30 variants.

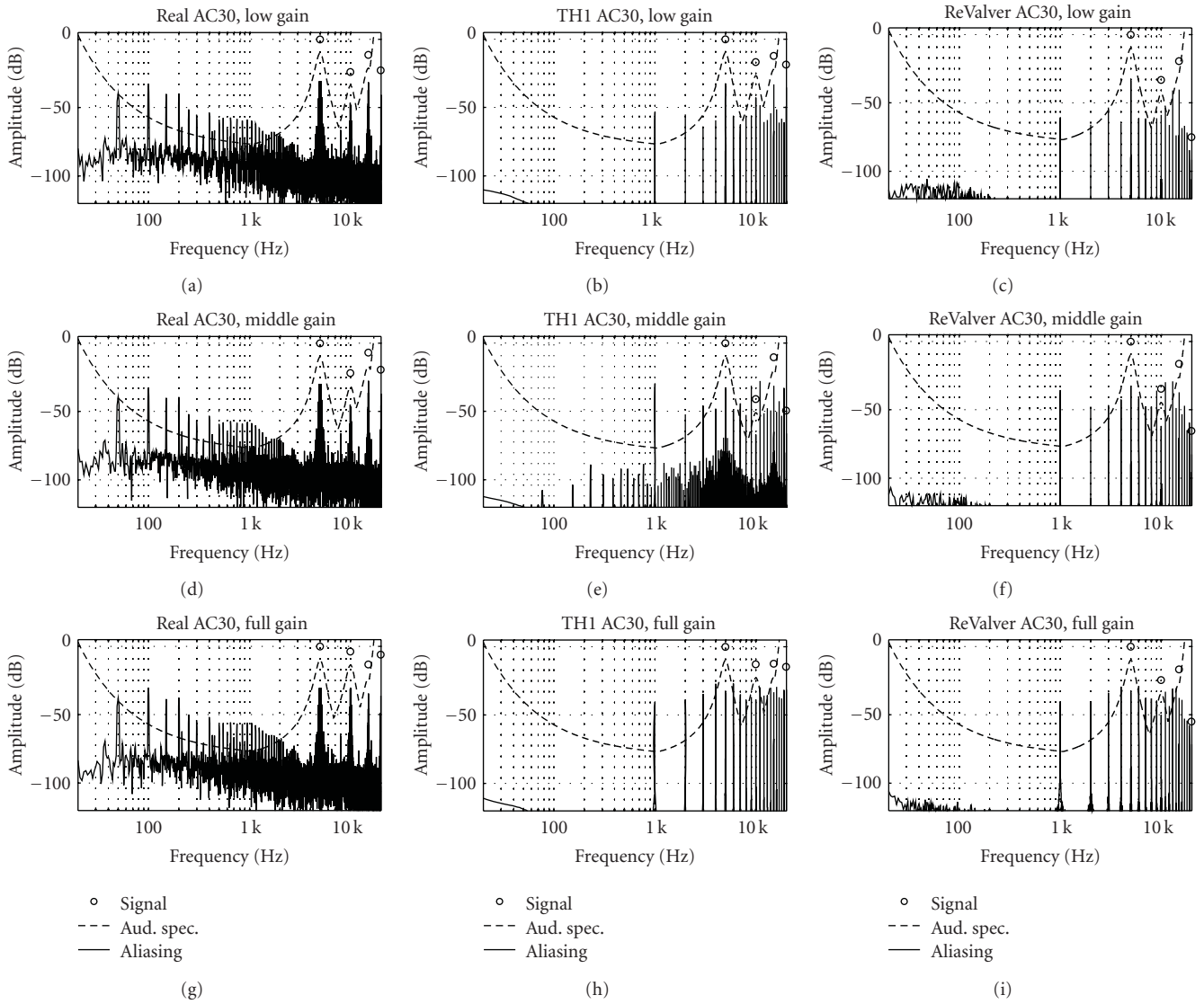


Figure 9: Aliasing analysis for the real AC30 (left column), the TH1 AC30 (middle column), and the ReValver AC30 (right column), when a 5 kHz sine is used as input. The gain knob was first set to 9 o'clock (top row), then to 12 o'clock (middle row), and finally to full (bottom row). In the figures, the circles denote the peaks of the harmonic components, while the solid line stands for aliased signal components and noise. The dashed line illustrates an auditory spectrum estimate. In both software plugins (middle and right column), the aliased components greatly exceed the estimated auditory threshold, thus suggesting that the aliasing is clearly audible. For the real AC30 (left column), the figures show that the 50 Hz power supply hum and its harmonics are also audible.

With full gain (bottom row), however, the intermodulation distortion figures reveal stronger differences. On the real AC30, the static IM distortion decreases with increasing input signal amplitude, while the DIM distortion remains nearly constant. The TH1 AC30 shows an interesting notch in the DIM distortion curve, indicating that there could be a local minimum of DIM distortion with input signals with a certain amplitude. The DIM/IM distortion on the ReValver AC30 with full gain seems to be relatively low and largely unaffected by the input signal amplitude.

In general, it seems that there are no large differences in the intermodulation distortion behavior between the real and simulated AC30 variants at low- and medium gain settings. However, the underlying mechanisms causing the different full-gain intermodulation distortion behavior between the AC30 variants should be examined in detail in further studies.

The fifth and final analysis plot is the aliasing analysis, depicted in Figure 9. It illustrates the estimated auditory spectrum (dashed line) caused by a distorted 5 kHz sine signal. The amplitude peaks of the distorted signal are plotted with circles. The solid lines in the figures illustrate the aliased signal components and measurement noise.

As can be seen in Figure 9, the real AC30 (first column) has a relatively high noise level due to the 50 Hz power supply hum and its harmonic components. This power supply

noise is clearly audible also in quiet parts when using the amplifier in a real playing situation. Interestingly, the aliasing analysis plots for both virtual AC30 plugins (second and third column) show severe aliasing phenomenon. The aliased component at 1 kHz is roughly 40 dB above the auditory spectrum estimate at full gain settings, making it clearly audible. In fact, the heavy aliasing behavior can strongly be heard also by listening to the logsweep responses of the AC30 plugins. It should be noted that the power supply noise is generally less irritating than aliasing noise, since the former stays largely constant regardless of the input signal. Aliasing noise, in turn, changes radically according to the input signal, and can it be an especially undesired phenomenon when playing high bent notes, since when the original tone moves up in frequency, the strongest aliased components move down, creating an unpleasant effect.

After all the analysis figures have been created, the DATK analysis part is finished. For the 2.4 GHz MacBook test computer, it took 76 seconds to analyze the AC30 responses (66 seconds when running under Matlab), while clearly most of the time is spent in performing the logsweep analysis.

4. Conclusion

A new software tool for performing rapid analysis measurements on nonlinearly distorting audio effects was presented. The software is called Distortion Analysis Toolkit (DATK), and it is freely available for download at http://www.acoustics.hut.fi/publications/papers/DATK/. The software operates by first creating an excitation signal as a.wav file according to the specifications set by the user. Next, the user feeds this excitation signal as an input to a physical or virtual distorting audio effect and records the output. Several recordings can be made if the distorting effect's control parameter variations are to be analyzed. Finally, the introduced software analyzes the response files and displays the analysis results with figures. The software is developed in Matlab language, but it can also be operated in standalone mode. The DATK includes five distortion analysis methods, and additional analysis techniques can be appended to it using Matlab.

The usage of the DATK was illustrated with a case study, where a custom-built AC30 guitar tube amplifier and two commercial software simulations, TH1 AC30 and ReValver AC30, were measured and compared. The amount and type of distortion on the real AC30 were found to be strongly dependent on the channel gain, as well as input signal frequency. Especially the amplitude ratios between the first few harmonic components show a complex nonmonotonous dependency from the channel gain. Although both software plugins successfully simulate the overall distortion characteristics in a broad sense, their response is more static than the real AC30 response. Waveform responses to a transient input signal reveal that the nonlinearity in the real AC30 is dynamic. Also the tested TH1 AC30 plugin shows strong dynamic behavior, while the ReValver AC30 plugin is more static. Intermodulation distortion analysis did not reveal any major differences between the three AC30 variants for low- and medium-gain settings, although some differences could be observed under full gain operation. Clearly audible aliasing noise was observed for both software plugins with sinusoid inputs when running a 48 kHz sample rate, whereas the real AC30 suffered from power supply hum. In a practical guitar playing condition, the aliasing noise on the tested software plugins is in most cases negligible due to the complex spectrum of the guitar. High bent notes can, however, reveal audible aliasing artifacts even when operating at a 96 kHz sample rate.

Acknowledgments

This research is funded by the Aalto University. The author is grateful to Vesa Välimäki for helpful comments. Special thanks are due to Ari Viitala for building the AC30 amplifier.

References

[1] J. Pakarinen and D. T. Yeh, "A review of digital techniques for modeling vacuum-tube guitar amplifiers," *Computer Music Journal*, vol. 33, no. 2, pp. 85–100, 2009.

[2] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Proceedings of the 108th AES Convention*, Paris, France, February 2000, preprint 5093.

[3] P. D. Hatziantoniou and J. N. Mourjopoulos, "Generalized fractional-octave smoothing of audio and acoustic responses," *Journal of the Audio Engineering Society*, vol. 48, no. 4, pp. 259–280, 2000.

[4] M. Otala, "Transient distortion in transistorized audio power amplifiers," *IEEE Transactions on Audio and Electroacoustics*, vol. 18, no. 3, pp. 234–239, 1970.

[5] E. Leinonen, M. Otala, and J. Curl, "A method for measuring transient intermodulation distortion (TIM)," *Journal of the Audio Engineering Society*, vol. 25, no. 4, pp. 170–177, 1977.

[6] V. Välimäki and A. Huovilainen, "Oscillator and filter algorithms for virtual analog synthesis," *Computer Music Journal*, vol. 30, no. 2, pp. 19–31, 2006.

[7] J. Pakarinen and M. Karjalainen, "Enhanced wave digital triode model for real-time tube amplifier emulation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 738–746, 2010.

[8] R. D. Patterson, "The sound of a sinusoid: spectral models," *Journal of the Acoustical Society of America*, vol. 96, no. 3, pp. 1409–1418, 1994.

[9] R. A. Wannamaker, "Psychoacoustically optimal noise shaping," *Journal of the Audio Engineering Society*, vol. 40, no. 7-8, pp. 611–620, 1992.

[10] K. Brandenburg, G. Stoll, Y. F. Dehéry, J. D. Johnston, L. V. D. Kerkhof, and E. F. Schröder, "The ISO-MPEG-1 audio: a generic standard for coding of high-quality digital audio," *Journal of the Audio Engineering Society*, vol. 42, no. 10, pp. 780–792, 1994.

[11] S. Van De Par, A. Kohlrausch, G. Charestan, and R. Heusdens, "A new psychoacoustical masking model for audio coding applications," in *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, vol. 2, pp. 1805–1808, Orlando, Fla, USA, May 2002.

[12] S. Errede, "Measurement of the electromagnetic properties of electric guitar pickups," Tech. Rep., February 2010, http://online.physics.uiuc.edu/courses/phys498pom/lab_handouts/electric_guitar_pickup_measurements.pdf.
[13] J. Elyea, "The history of Vox," April 2010, http://www.voxamps.com/history.
[14] R. Aiken, "Is the Vox AC-30 really class A?" 2002, http://www.aikenamps.com/VoxAC30classA_2.html.
[15] "Overloud TH1," February 2010, http://www.overloud.com/.
[16] "Peavey ReValver," February 2010, http://www.peavey.com/products/revalver/index.cfm.
[17] "Cockos Reaper," February 2010, http://www.cockos.com/reaper.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 793816, 8 pages
doi:10.1155/2010/793816

Review Article

Analysis, Synthesis, and Classification of Nonlinear Systems Using Synchronized Swept-Sine Method for Audio Effects

Antonin Novak, Laurent Simon, and Pierrick Lotton

Laboratoire d'Acoustique, Université du Maine, UMR CNRS 6613, 72000 Le Mans, France

Correspondence should be addressed to Antonin Novak, ant.novak@gmail.com

Received 1 March 2010; Revised 16 June 2010; Accepted 5 July 2010

Academic Editor: Mark Sandler

A new method of identification, based on an input synchronized exponential swept-sine signal, is used to analyze and synthesize nonlinear audio systems like overdrive pedals for guitar. Two different pedals are studied; the first one exhibiting a strong influence of the input signal level on its input/output law and the second one exhibiting a weak influence of this input signal level. The Synchronized Swept Sine method leads to a Generalized Polynomial Hammerstein model equivalent to the pedals under test. The behaviors of both pedals are illustrated through model-based resynthesized signals. Moreover, it is also shown that this method leads to a criterion allowing the classification of the nonlinear systems under test, according to the influence of the input signal levels on their input/output law.

1. Introduction

Various classical analog audio effects fall into the category of nonlinear effects such as compression, harmonic excitation, overdrive, or distortion for guitars. Digital emulations of nonlinear audio effects can be obtained when using a suitable nonlinear model. Such nonlinear models are available in the literature, for example, Volterra model [1], neural network model [2], MISO model [3], NARMAX model [4], hybrid genetic algorithm [5], extended Kalman filtering [6], or particle filtering [7].

A new method for the identification of nonlinear systems, based on an input exponential swept-sine signal has been proposed by Farina et al. [8, 9]. This method has been recently modified for the purpose of nonlinear model estimation [10] and allows a robust and fast one-path analysis and identification of the unknown nonlinear system under test. The method is called Synchronized Swept Sine method as it uses a synchronized swept sine signal for identification.

A nonlinear effect can be modeled either by a simple static nonlinear input/output law, where each input amplitude is directly mapped to an output amplitude (nonlinear system without memory), or on a more complex way by nonlinear laws which take memory into account, meaning that the memoryless nonlinearities and the linear filtering are mixed. Moreover, several nonlinear audio effects include amplifiers, the gain of which is automatically controlled by the level of the input signal [11]. In other words, the performance of nonlinear systems with memory may also depend on parameters of the input signal, such as its level or its past extrema, as for the hysteretic systems [12].

This classification of nonlinear systems according to the influence of the input signal parameters on the input/output law leads to a similar classification of the identification methods. The methods for identification of static nonlinearities indeed do not require the same level of model complexity as methods used for nonlinear systems with memory or with gain control.

In this paper, it is shown that the Synchronized Swept Sine method is suited to analyze, classify, and synthesize the nonlinear systems under test. In the frame of this work, two different overdrive pedals have been tested; the first one exhibiting a strong influence of the input signal level on its input/output law and the second one exhibiting a weak influence of this input signal level.

In Section 2, Synchronized Swept Sine method is shortly presented. This method leads to a nonlinear model (Section 3), made up of several branches, each branch consisting of a nonlinear function and a linear filter. The nonlinear functions are chosen as a power series that makes the model equivalent to a Generalized Polynomial Hammerstein (GPH) model. Next, the measurements on overdrive pedals are presented in Section 4. The behaviors of both systems are illustrated through model-based resynthesized signals. Finally, in Section 5, we propose a criterion based on the GPH model to classify the nonlinear systems according to the importance of the influence of the input signal parameters on the input/output law of the system under test.

2. Analysis of Nonlinear Systems

The nonlinear system identification method used in this paper is based on an excitation by a swept-sine signal (also called chirp) exhibiting an exponential instantaneous frequency $f_i(t)$. This so-called Synchronized Swept-Sine method allows the identification of a system in terms of harmonic distortion at several orders. This identification is conducted in several steps.

First, an exponential swept-sine signal $x_s(t)$ is generated and used as the input signal of the nonlinear system under test. The excitation swept-sine signal $x_s(t)$ is defined as

$$x_s(t) = A_s \sin\left\{2\pi L\left[\exp\left(\frac{f_1 t}{L}\right) - 1\right]\right\}, \tag{1}$$

where

$$L = \text{Round}\left(\frac{\hat{T} f_1}{\ln(f_2/f_1)}\right), \tag{2}$$

f_1 and f_2 being start and stop frequencies, and $\hat{T}$ being the time length of the swept-sine signal. The rounding operator is necessary to synchronize the swept-sine signal for higher-order contributions with linear component as depicted in Figure 1. This condition is necessary for the model identification and for a proper reconstruction of the output signal.

Then, the distorted output signal $y_s(t)$ of the nonlinear system is recorded for use in the so-called nonlinear convolution [8]. Next, the signal denoted $\widetilde{x}_s(t)$ is derived from the input signal $x_s(t)$ as its time-reversed replica with amplitude modulation in such a way that the convolution between $x_s(t)$ and $\widetilde{x}_s(t)$ gives a Dirac delta function $\delta(t)$. The signal $\widetilde{x}_s(t)$ is called the inverse filter [8].

Finally, the convolution between the output signal $y_s(t)$ and the inverse filter $\widetilde{x}_s(t)$ is performed, leading to

$$y_s(t) * \widetilde{x}_s(t) = \sum_{i=1}^{\infty} h_i(t + \Delta t_i), \tag{3}$$

where $h_i(t)$ are the higher-order impulse responses and Δt_i are the time lags between the first and the ith impulse response. Since the result of convolution $y_s(t) * \widetilde{x}_s(t)$ consists of a set of higher-order impulse responses that are time shifted, each partial impulse response can be separated from each other.

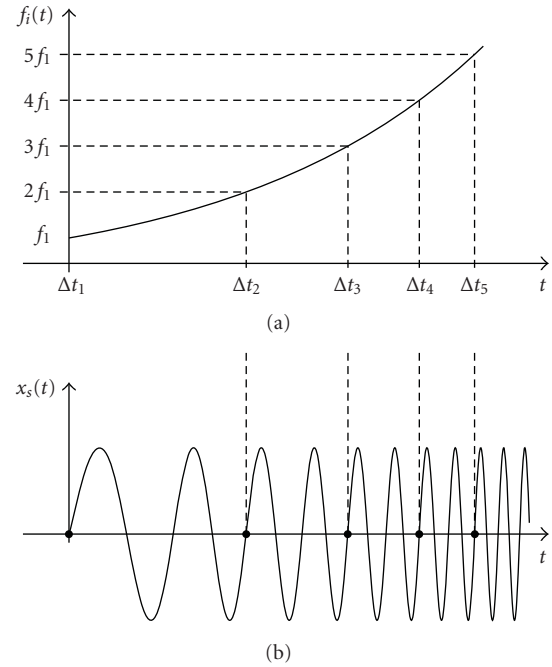


Figure 1: Swept-sine signal $x_s(t)$ in the time domain (b), with the time length chosen according to the instantaneous frequency $f_i(t)$ (a).

The set of higher-order nonlinear impulse responses $h_i(t)$ can also be expressed in the frequency domain. The frequency response functions of the higher-order nonlinear impulse responses $h_i(t)$ are then defined as their Fourier transforms

$$H_i(f) = \mathbf{FT}[h_i(t)]. \tag{4}$$

The frequency responses $H_i(f)$ represent the frequency dependency of the higher-order components. $H_i(f)$ may be regarded as the system frequency response, when considering only the effect of the input frequency f on the ith harmonic frequency of the output. The theoretical background of the Synchronized Swept-Sine method is detailed in [10].

3. Model Identification

In this section, the frequency responses $H_i(f)$ described in the previous section are used for a nonlinear model based on a multiple-input single-output (MISO) model [3]. The structure of this model is shown in Figure 2. It is made up of N parallel branches, each branch consisting of a linear filter $A_n(f)$. The input signals $g_n[x(t)]$ are known as linear and/or nonlinear functions of $x(t)$ chosen by the user.

The output signal $y(t)$ of the nonlinear system can then be expressed as

$$y(t) = \sum_{n=1}^{N} \int_{-\infty}^{\infty} g_n[x(\tau)] a_n(t - \tau) d\tau, \tag{5}$$

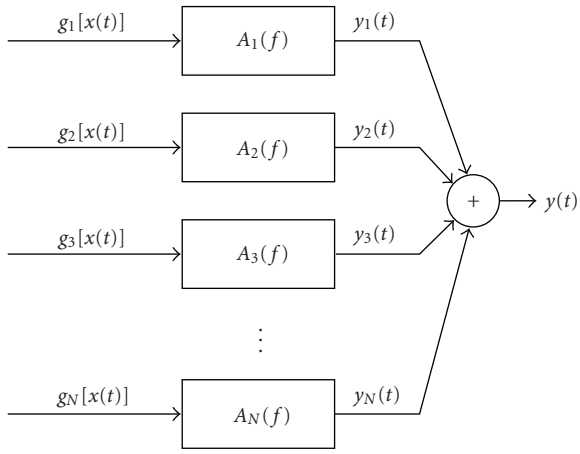


Figure 2: MISO model for nonlinear system identification with the input signals $g_n[x(t)]$ and the linear filters $A_n(f)$, $n \in [1, N]$.

where N is the number of the input signals of the MISO-based nonlinear model, and where $a_n(t)$ is the impulse response related to the nth branch of the MISO based nonlinear model

$$a_n(t) = \mathbf{FT}^{-1}[A_n(f)]. \tag{6}$$

The linear filters $A_n(f)$ (or equivalently the impulse responses $a_n(t)$) have then to be identified, using the previously estimated $H_i(f)$. This identification consists in solving a linear system of N equations using the least-squares method. First, the coefficients $c_{n,k}$ of Discrete Fourier Series of the functions $g_n[x(t)]$ are calculated as

$$c_{n,k} = \frac{2}{M}\sum_{m=0}^{M-1} g_n\left[\sin\left(\frac{2\pi}{M}m\right)\right]\exp\left(-j\frac{2\pi}{M}km\right), \tag{7}$$

for an input signal being a discrete-time harmonic signal of length M. Next, the following set of linear equations with unknown $A_n(f)$ is solved

$$H_i(f) = \sum_{n=1}^{N} A_n(f)c_{n,i} + \mathrm{Res}(f), \tag{8}$$

for $i \in [1, I]$ (I being the number of harmonics taken into account), $n \in [1, N]$, and $\mathrm{Res}(f)$ being the residue. As $I \geq N$, there can be more equations than unknowns. To solve the set of equations (8) for $I > N$, the least-squares algorithm [13] is applied, minimizing the residue $\mathrm{Res}(f)$.

If the functions $g_n[x(t)]$ are improperly chosen and/or if at least one of the input signals is missing, the value of the residue increases drastically, which makes $\mathrm{Res}(f)$ an *a posteriori* criterion for the choice of the input signals $g_n[x(t)]$.

If one of the nonlinear functions $g_n[x(t)]$ produces high harmonic distortion components, nonlinear aliasing [14] can appear. This can be avoided by choosing the nonlinear functions $g_n[x(t)]$ according to any mathematical series. The most used series is the one based on the power series, such as

$$g_n[x(t)] = x^n(t). \tag{9}$$

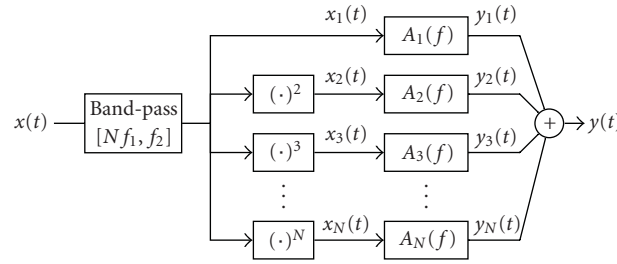


Figure 3: Generalized Polynomial Hammerstein (GPH) model (power series nonlinear model) for nonlinear system identification.

A model with inputs chosen as power series is equivalent to the Generalized Polynomial Hammerstein (GPH) model [15] with N branches. In such a case, the nonlinear aliasing can be controlled by the frequency range. The highest frequency must not exceed $f_s/(2N)$, where f_s is the sampling frequency. The lowest frequency limit is as well given by the highest power function N. The filters $A_n(f)$ are indeed valid only in the frequency band $[Nf_1, f_2]$. For that reason, the model should be preceded by a bandpass filter as shown in Figure 3. The amplitude limitation is as well given by the excitation signal $x_s(t)$ used for the analysis. As the nonlinear system is tested using an excitation signal $x_s(t)$, the level of which does not exceed the amplitude A_s, the nonlinear system is valid only for an input signal not exceeding A_s.

4. Experimental Measurements: Analysis and Synthesis

In a previous work, the Synchronized Swept-Sine method has been used to model the limiter part of a dynamic processor [10]. Results have shown the ability of the method to estimate very hard distortions with a good accuracy within the whole frequency range. In this section, the same method is tested on two real-world analog audio effects devices exhibiting weak distortions. Both devices under test are overdrive effect pedals. The first one is an Ibanez Tube Screamer ST-9 [16], the second one is a home-made overdrive pedal, the electric circuit diagram of which being depicted in Figure 4. These pedals exhibit different nonlinear performances, as investigated below.

The experimental measurement consists of two steps: (a) identification of the nonlinear system under test through the GPH model as described in the previous section and (b) comparison of the output signals of both the nonlinear system under test and the GPH model when excited with the same signal.

For the first step, the measurement setup is as follows: the sampling frequency used for the experiment is $f_s = 192\,\mathrm{kHz}$ and the excitation signal $x_s(t)$ is sweeping from $f_1 = 5\,\mathrm{Hz}$ to $f_2 = 10\,\mathrm{kHz}$ with a maximum amplitude $A_s = 1\,\mathrm{V}$. The filters $A_n(f)$ of the GPH model are then estimated.

The second step is the validation of the model for several input levels. To analyze the accuracy of the GPH model, the following test is performed. An input signal is provided to the inputs of both the real-world analog effect device and its

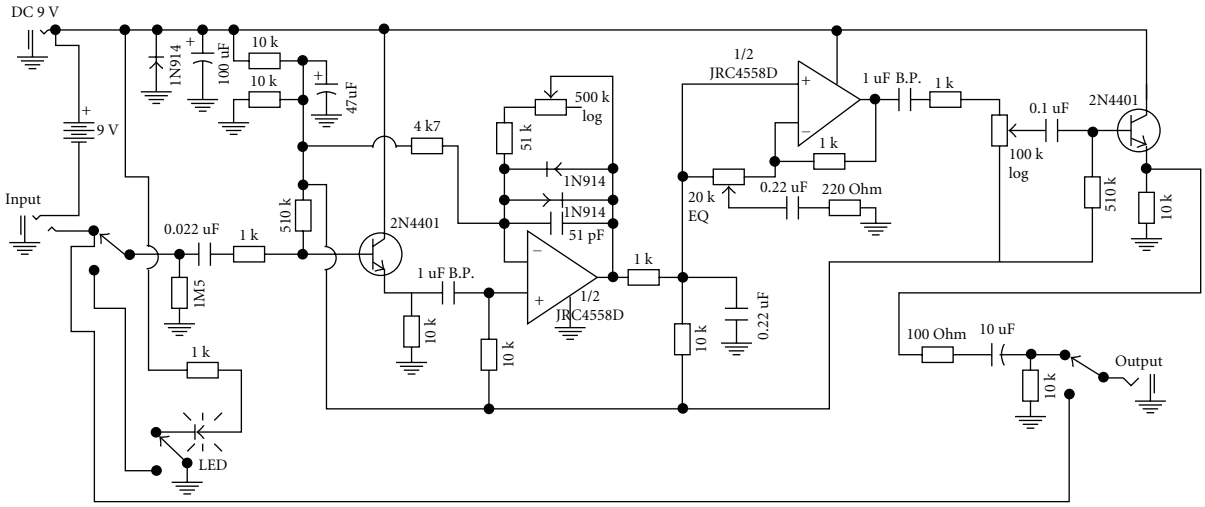


Figure 4: Circuit diagram of the home-made overdrive effect pedal.

corresponding GPH model, and both outputs are compared in the time and frequency domains. The input signal $x(t)$ is a sine-wave with frequency $f_0 = 500\,\text{Hz}$ and amplitude A_0 that varies from 0.1 V to 1 V with step 0.1 V. Regarding distortion measurements, we choose to test the accuracy of the method through the weighted harmonic distortion (HI-2) that takes into account the higher-order components more than the classical harmonic distortion [17].

4.1. Computing Complexity versus Accuracy. The choice of the number of branches N of the GPH model is a key parameter which may influence the accuracy of the identification. The higher the value of N the higher the accuracy but the higher the computing complexity. To choose an optimal value of N, the Ibanez Tube Screamer is firstly tested for different values of N. Then, the HI-2 is calculated for both output signals, the output of the real-world system and the GPH model-based output, when excited with a sine wave with frequency $f_0 = 500\,\text{Hz}$ and amplitude $A_0 = 1\,\text{V}$.

Both the HI-2 difference, noted ΔHI-2 and given in dB, and the relative computing complexity CC are presented in Table 1. The CC is defined as the computational time needed to generate the output of the GPH model with N branches, normalized by the computational time needed to generate the output of a GPH model for $N = 1$ (linear system case). (The simulation is made in Matlab for $f_s = 192\,\text{kHz}$ and for a signal with a number of samples equal to f_s.)

As shown in Table 1 (for the nonlinear system under test), the choice $N = 7$ is a good candidate for an optimal value between the accuracy and the computational time. Increasing N does indeed not increase the accuracy of the model, but increases the computational time. As the model is made up of parallel branches (each branch including the same computing complexity), the computational time is directly proportional to N.

4.2. Ibanez Tube Screamer Overdrive Pedal. The first nonlinear system under test is an overdrive effect pedal Ibanez Tube Screamer [16] (pedal 1). The pedal has been configured as follows: Drive = 4/10, Level = 6/10, Mid Boost = 0/10, and Tone = 3/10. Driving input level is attenuated by 25 dB before exciting the nonlinear system under test.

The outputs corresponding to an input sine wave with $f_0 = 500\,\text{Hz}$ and $A_0 = 1\,\text{V}$ are shown in Figure 5, in both time and frequency domains. The HI-2 is $-21.14\,\text{dB}$ for the real-world output and $-21.17\,\text{dB}$ for the model output ($N = 7$), that illustrates a very good accuracy of the identification method.

The HI-2 for both model and real-world system are compared in Figure 6, when measured with increasing input signal level A_0 (from 0.1 V to 1 V). The HI-2 fits only for the maximum input level $A_0 = 1\,\text{V}$, corresponding to the level A_s of the signal used for the estimation of the $A_n(f)$. For other levels $A_0 < 1\,\text{V}$, discrepancies between the HI-2 spread from 4 to 8 dB. For example, the responses of the GPH model and the real-world device to a sine wave, the amplitude of which is $A_0 = 0.5\,\text{V}$, is given in Figure 7, in both time and frequency domains. The regenerated output signal, based on the GPH model estimated for $A_s = 1\,\text{V}$, does not fit with the real-world output signal. The HI-2 is $-28.0\,\text{dB}$ for the real-world output and $-34.6\,\text{dB}$ for the model output.

As a consequence, the nonlinear system under test (Ibanez Tube Screamer) can be seen as a nonlinear system whose input/output law is driven by the input level A_0. Nevertheless, when the amplitude A_0 of the input signal is the same as the amplitude A_s used for the identification of the nonlinear system, the GPH model-based output fits well with the real-world output. This is illustrated in Figure 8 for the case $A_s = A_0 = 0.5\,\text{V}$. The HI-2 is then $-28.0\,\text{dB}$ for the real-world output and $-27.9\,\text{dB}$ for the GPH model-based output.

4.3. Home-Made Overdrive Pedal. The second nonlinear system under test is a home-made overdrive pedal, noted pedal 2, exhibiting lower dependency on input level. The circuit diagram of the pedal 2 is presented in Figure 4.

Table 1: Effect of the number of branches N of the GPH model on the weighted harmonic distortion difference ΔHI-2 and on the relative computing complexity CC.

N	1	2	3	4	5	6	7	8	9	10	11
ΔHI-2 [dB]	$-\infty$	39.8	1.9	1.9	0.31	0.31	0.03	0.03	0.03	0.03	0.03
CC [−]	1.0	2.0	3.1	4.2	5.4	6.5	7.5	8.6	9.7	10.7	11.9

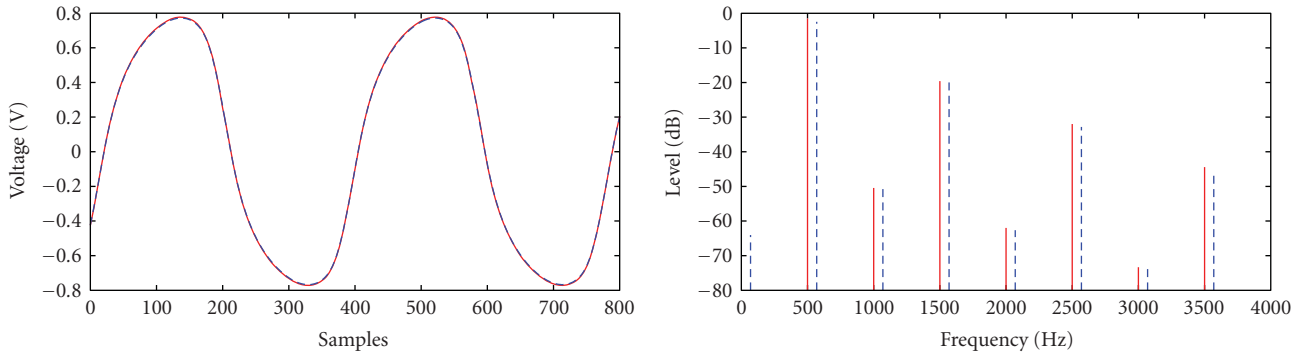


Figure 5: Comparison between the real-world output of pedal 1 (blue-dashed) and the GPH model-based output (red-solid) in time (a) and frequency (b) domains, for a sine wave excitation with $f_0 = 500$ Hz and $A_0 = 1$ V. The GPH model is estimated using a swept sine signal with amplitude $A_s = 1$ V. For the sake of clarity, the output of the real-world device in the frequency domain is shifted to the right.

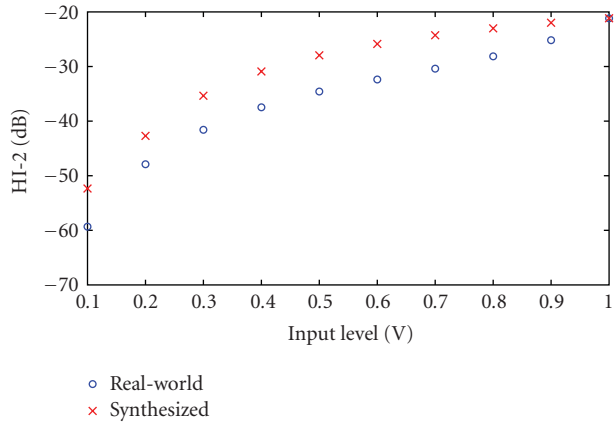


Figure 6: HI-2 of synthesized and real-world signals as a function of the input level (pedal 1).

The same configuration and analysis as those described in Section 4.2 have been setup.

The outputs corresponding to an input sine wave of $f_0 = 500$ Hz and $A_0 = 1$ V are shown in Figure 9. The HI-2 is −29.56 dB for the real-world output and −29.45 dB for the GPH model output. As for the case of pedal 1, it illustrates a very good accuracy of the identification method. The outputs corresponding to an input sine wave of $f_0 = 500$ Hz and $A_0 = 0.5$ V (Figure 10) show also a good agreement even if the amplitude A_0 of the input signal differs of the amplitude $A_s = 1$ V used for the identification of the nonlinear system. The HI-2 is −45.0 dB for the real-world output and −43.2 dB for the model output.

As illustrated in Figure 11, the difference ΔHI-2 between both HI-2 is less than 2.5 dB for all the input levels A_0. Thus, such a nonlinear system represents a system whose input/output law is not driven by the input level A_0. For such a nonlinear system, the presented identification method with GPH model can be used for both analysis and synthesis.

5. Classification of Input Level (In)Dependent Nonlinear Systems

In the previous section, two real-world nonlinear systems, exhibiting different nonlinear behaviors have been identified thanks to the Synchronized Swept Sine method. The input/output law of the first system under study (pedal 1) is driven by the input level A_0, while the input/output law of the second one (pedal 2) is independent of this input level. In the following, we call "input level dependent" the first kind of nonlinear system and "input level independent" the second one.

A key point of the method of identification presented in this paper is its capacity to distinguish both kinds of nonlinear systems through its ability to synthesize the output signals from any given input signal. Then, the classification of nonlinear systems in these two categories (input level dependent and input level independent) is performed here thanks to the Synchronized Swept-Sine method. A criterion based on the analysis of impulse responses $a_n(t)$ of GPH model is used to perform this classification.

More specifically, we show that analyzing only the first branch (linear part) of the model is sufficient to classify both kinds of nonlinear systems. The linear impulse response $a_1(t)$ is firstly estimated for $N = 7$ and for several input levels $A_s \in [0.1, 1]$ V, noted $a_{1,l}(t)$; l denoting the input of the index

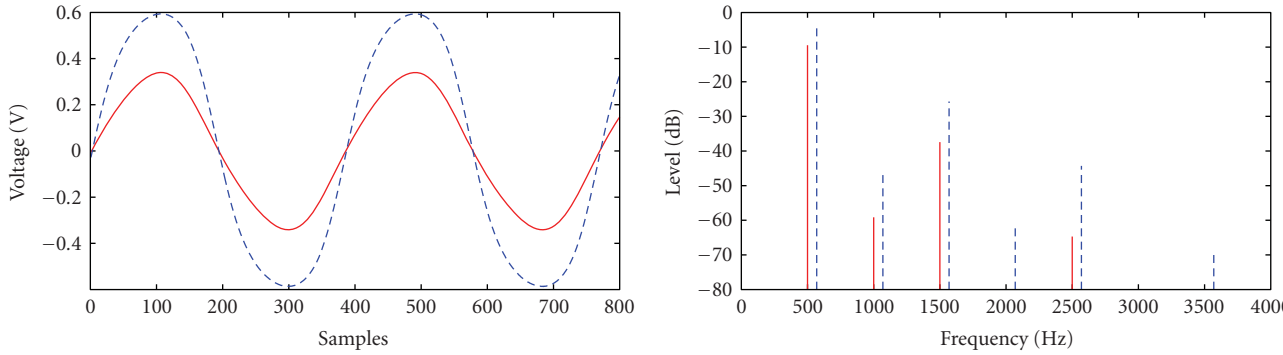


Figure 7: Comparison between the real-world output of pedal 1 (blue-dashed) and the GPH model-based output (red-solid) in time (a) and frequency (b) domains, for a sine wave excitation with $f_0 = 500$ Hz and $A_0 = 0.5$ V. The GPH model is estimated using a swept sine signal with amplitude $A_s = 1$ V. For the sake of clarity, the output of the real-world device in the frequency domain is shifted to the right.

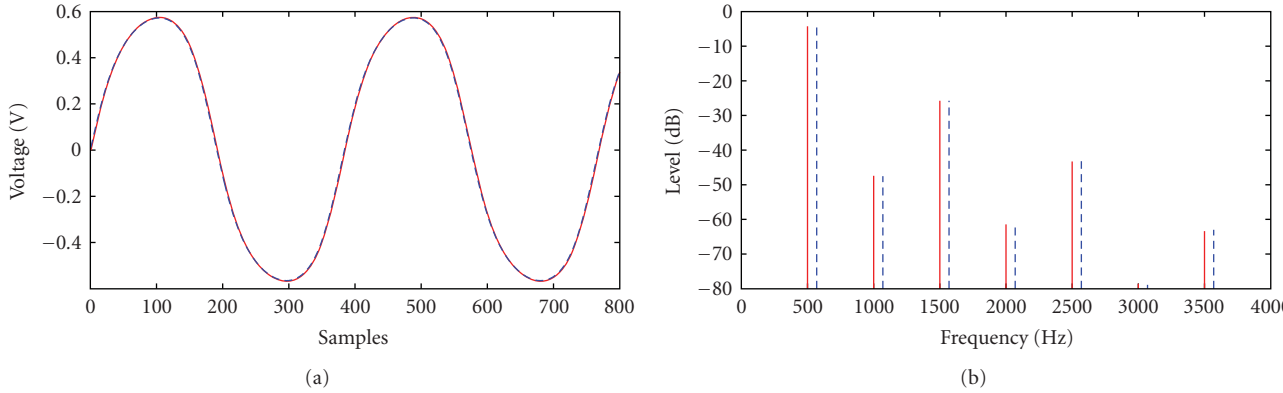


Figure 8: Comparison between the real-world output of pedal 1 (blue-dashed) and the GPH model-based output (red-solid) in time (a) and frequency (b) domains, for a sine wave excitation with $f_0 = 500$ Hz and $A_0 = 0.5$ V. The GPH model is estimated using a swept sine signal with amplitude $A_s = 0.5$ V. For the sake of clarity, the output of the real-world device in the frequency domain is shifted to the right.

level. Then, if the nonlinear system under test is an "input level dependent" one, the impulse responses are expected to be different from each other. On the contrary, if the nonlinear system under test is an "input level independent" one, the impulse responses $a_{1,l}(t)$ are expected to be very close each other.

In Figures 12 and 13, the impulse responses $a_{1,l}(t)$ of the first branch of the nonlinear model are depicted for different input levels, for the case of pedals 1 and 2 respectively. Using these results, we propose to define the following relative squared error (RSE) based criterion for classifying the nonlinear systems under test,

$$\mathrm{RSE}_l = \frac{\int \left\{a_{1,l}(t) - \langle a_1(t)\rangle\right\}^2 dt}{\int \langle a_1(t)\rangle^2 dt}, \tag{10}$$

where $\langle a_1(t)\rangle$ is the average impulse response,

$$\langle a_1(t)\rangle = \frac{\sum_{l=1}^{10} a_{1,l}(t)}{10}. \tag{11}$$

The RSE measures the mean-squared distance between the average impulse response $\langle a_1(t)\rangle$ and the impulse responses $a_{1,l}(t)$.

For the case of pedal 1, we have $\max(\mathrm{RSE}_l) = 10\%$, whilst for the case of pedal 2, $\max(\mathrm{RSE}_l) = 1.3\%$. This order of magnitude between both values clearly allows to classify the input level dependent and input level independent nonlinear systems under test.

6. Conclusions

In this paper, a recently proposed method [10] is tested for classifying, analyzing, and synthesizing two nonlinear systems (overdrive effect pedals) exhibiting different nonlinear behaviors. The method for identification of nonlinear systems is based on synchronized swept-sine signal and allows the identification of nonlinear system under test in a one-path measurement.

The classification is indispensable for distinguishing nonlinear systems whose input/output law is driven by input level, and nonlinear systems whose input/output law is independent of the input level.

Two nonlinear systems have been tested: the first one corresponding to a nonlinear system whose input/output law is driven by the input level and the second being

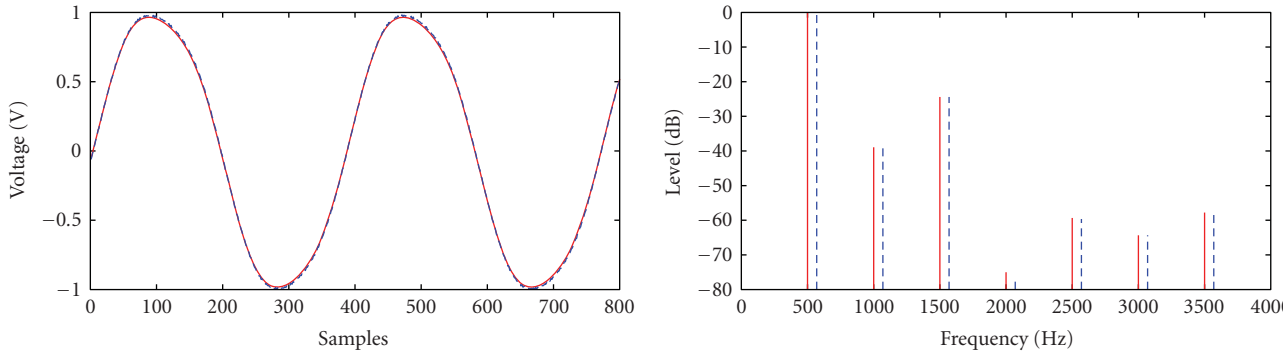


FIGURE 9: Comparison between the real-world output of pedal 2 (blue-dashed) and the GPH model-based output (red-solid) in time (a) and frequency (b) domains, for a sine wave excitation with $f_0 = 500$ Hz and $A_0 = 1$ V. The GPH model is estimated using a swept sine signal with amplitude $A_s = 1$ V. For the sake of clarity, the output of the real-world device in the frequency domain is shifted to the right.

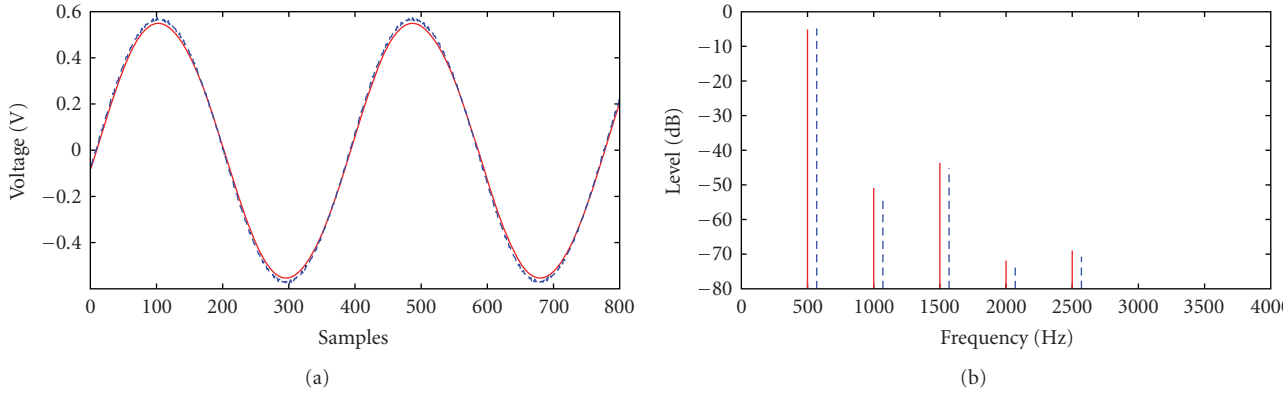


FIGURE 10: Comparison between the real-world output of pedal 2 (blue-dashed) and the GPH model-based output (red-solid) in time (a) and frequency (b) domains, for a sine wave excitation with $f_0 = 500$ Hz and $A_0 = 0.5$ V. The GPH model is estimated using a swept sine signal with amplitude $A_s = 1$ V. For the sake of clarity, the output of the real-world device in the frequency domain is shifted to the right.

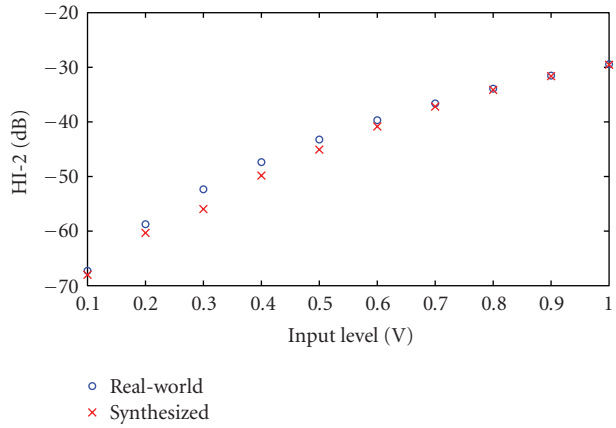


FIGURE 11: HI-2 of synthesized and real-world signals as a function of the input level (pedal 2).

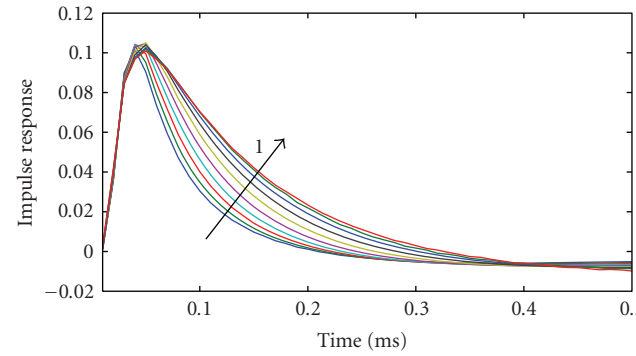


FIGURE 12: Impulse responses $a_{1,l}(t)$ of the first branch of the nonlinear model ($N = 7$) depicted for different input levels (pedal 1).

a nonlinear system whose input/output law is independent of the input level. For the latter (pedal 2), the results show that the method is useful for both analysis and synthesis. The comparison between the synthesized and real-world signal shows very good agreement in both time and frequency domains. The same agreement is shown by comparing the weighted harmonic distortion HI-2 [17].

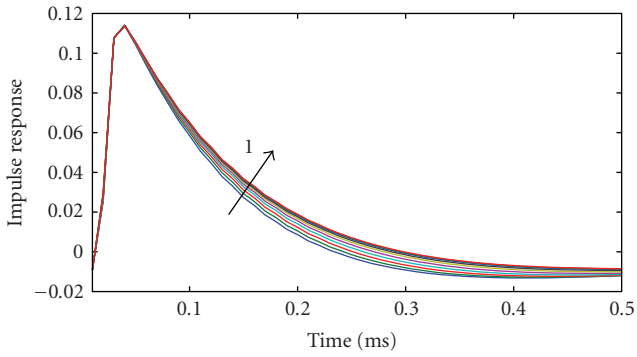


FIGURE 13: Impulse responses $a_{1,l}(t)$ of the first branch of the nonlinear model ($N = 7$) depicted for different input levels (pedal 2).

In the case of input level dependent nonlinear system (pedal 1), it is shown that when the identification is carried out from a signal with input level A_s, the model is very accurate only when the amplitude of the input signal to be synthesized is $A_0 = A_s$. Thus, for a whole analysis of such a system, the frequency responses $H_i(f)$ have then to be estimated for different input levels A_s, leading to 2D frequency response functions (FRF) $H_i(f, A_s)$.

Works are now in progress to implement the FRF $H_i(f, A_s)$ into the nonlinear model in order to synthesize such systems for any input signals.

Acknowledgments

This work was supported by the French region *Pays de la Loire*. The authors would like to thank J. B. Doc and J. C. Le Roux for their help in selecting two overdrive pedals, and to the anonymous reviewers for their helpful comments.

References

[1] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, John Wiley & Sons, New York, NY, USA, 1980.

[2] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, Springer, Berlin, Germany, 2001.

[3] H. Kusaka, M. Kominami, K. Matsumoto, and Y. Tashiro, "Performance improvement of CDMA/QPSK systems with nonlinear channel using decision feedback equalizers," in *Proceedings of the 5th IEEE International Symposium on Spread Spectrum Techniques & Applications*, pp. 812–817, September 1998.

[4] F. Thouverez and L. Jezequel, "Identification of NARMAX models on a modal base," *Journal of Sound and Vibration*, vol. 189, no. 2, pp. 193–213, 1996.

[5] Y.-W. Chen, S. Narieda, and K. Yamashita, "Blind nonlinear system identification based on a constrained hybrid genetic algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 3, pp. 898–902, 2003.

[6] H. W. Sorenson, *Kalman Filtering: Theory and Application*, IEEE Press, Montvale, NJ, USA, 1985.

[7] O. Cappe, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[8] A. Farina, A. Bellini, and E. Armelloni, "Non-linear convolution: a new approach for the auralization of distorting systems," in *Proceedings of the 110th Audio Engineering Society Convention*, Amsterdam, The Netherlands, May 2001.

[9] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Proceedings of the 108th Audio Engineering Society Convention*, Paris, France, February 2000.

[10] A. Novák, L. Simon, F. Kadlec, and P. Lotton, "Nonlinear system identification using exponential swept-sine signal," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2220–2229, 2010.

[11] U. Zölzer, *DAFX: Digital Audio Effects*, John Wiley & Sons, New York, NY, USA, 2002.

[12] G. Bertotti and I. D. Mayergoyz, *The Science of Hysteresis*, Academic Press, Boston, Mass, USA, 2006.

[13] G. Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Cambridge, Mass, USA, 1986.

[14] Y. Zhu, "Generalized sampling theorem," *IEEE Transactions on Circuits and Systems II*, vol. 39, no. 8, pp. 587–588, 1992.

[15] A. Janczak, *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models: A Block-Oriented Approach*, Springer, Berlin, Germany, 2005.

[16] Ibanez ST 9, "Super Tube Screamer," 2007, http://www.ibanez.com/Electronics/model-TS9DX.

[17] W. Klippel, "Measurement of Weighted Harmonic Distortion HI-2," in *Application Note*, Klippel GmbH, 2002.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 465417, 9 pages
doi:10.1155/2010/465417

Research Article

Automatic Noise Gate Settings for Drum Recordings Containing Bleed from Secondary Sources

Michael Terrell, Joshua D. Reiss, and Mark Sandler

The Centre for Digital Music, School of Electronic Engineering and Computer Science, Queen Mary University of London, London E14NS, UK

Correspondence should be addressed to Michael Terrell, michael.terrell@eecs.qmul.ac.uk

Received 1 March 2010; Revised 9 September 2010; Accepted 31 December 2010

Academic Editor: Augusto Sarti

An algorithm is presented which automatically sets the attack, release, threshold, and hold parameters of a noise gate applied to drum recordings which contain bleed from secondary sources. The gain parameter which controls the amount of attenuation applied when the gate is closed is retained, to allow the user to control the strength of the gate. The gate settings are found by minimising the artifacts introduced to the desirable component of the signal, whilst ensuring that the level of bleed is reduced by a certain amount. The algorithm is tested on kick drum recordings which contain bleed from hi-hats, snare drum, cymbals, and tom toms.

1. Introduction

Dynamic audio effects apply a control gain to the input signal. The gain applied is a nonlinear function of the level of the input signal (or a secondary signal). Dynamic effects are used to modify the amplitude envelope of a signal. They either compress or expand the dynamic range of a signal. A noise gate is an extreme expander. If the level of the signal entering the gate is below the gate threshold, an attenuation is applied. If the level of the signal is above the threshold the signal passes through unattenuated. The attack and release parameters control how quickly the gate opens and closes. As the name suggests, noise gates are used to reduce the level of noise in a signal. There are many audio applications, for example, noise gates are used to remove, breathing from vocal tracks, hum from distorted guitars, and bleed on drum tracks, particularly snare and kick drum tracks. The use of digital audio workstations (DAWs) for postproduction means that it is quick and easy to manually remove some sources of noise by silencing regions of an audio file. However, it is very time consuming to manually remove bleed from drum tracks so noise gates are still heavily used.

The reader is referred to [1] for a comprehensive review of digital audio effects (DAFx). In [2], a class of sound transformations called adaptive digital audio effects (A-DAFx) are defined. Adaptive effects extract features from a signal and use them to derive control parameters for sound transformations. Adaptive audio effects have existed for many years. Dynamic effects are simple examples of A-DAFx because the control gain applied is derived from the level of the input signal. Features can be extracted from the input signal, an external signal, or the output signal before being mapped to control parameters. These are referred to as *autoadaptive*, *external-adaptive,* and *feedback-adaptive* respectively. Cross-adaptive effects use two or more inputs; the features of which are used in combination to produce the control parameters for the sound transformation.

A-DAFx have been used for automatic mixing applications. Early work focused on audio for conferencing. An adaptive threshold gate is presented in [3]. This is an external adaptive effect. Ambient noise is picked up by a secondary microphone from which the level is extracted. The level of the noise is mapped to the threshold of a noise gate which is applied to the primary microphone. In [4], a direction sensitive gate is presented. This is a cross-adaptive effect. Each microphone unit contains two microphones. These face toward and away from the speaker. The level of the signals entering the microphones is extracted and

compared to determine the direction of the signal. The direction is mapped to an on/off switch which ensures that the microphone is only active if the sound source is in front of it.

Recent automatic mixing work has turned toward audio production. Perez-Gonzalez and Reiss [5–7] have presented A-DAFx for live audio production. A cross-adaptive effect which does automatic panning is presented in [5]. The automatic panner extracts spectral features from a number of channels, each of which corresponds to a different instrument. The spectral features are mapped to panning controls, subject to predefined priority rules. The objective is to separate spatially those instruments with similar frequency content. The work in [6] is used to reduce spectral masking of a target channel in a multichannel setup. This is a cross-adaptive effect. It extracts spectral features from each channel, and if a channel has a similar spectral content to the predefined target channel an attenuation is applied. Automatic fader control is demonstrated in [7]. This is a cross-adaptive effect. It extracts the loudness from each channel. Loudness is a perceptual feature, a function of level and spectral content. The loudness of each channel is compared to the average loudness of all channels and is mapped to fader controls. This mapping seeks to make the loudness of all channels equal.

In [7] the cross-adaptive effect is used to instantiate changes to the fader controls which seek to produce a predefined outcome: equal loudness in all channels. This can be viewed as a form of real-time optimization. There are a few examples of audio effect parameter automation, where the optimization is performed offline. Whilst these do not fit neatly into the A-DAFx structure, they still incorporate feature extraction and feature mapping. In [8], a method is presented which allows perceptual changes in equalization to be made to an audio signal. An example requirement is to make the signal sound brighter. This is a cross-adaptive effect. The spectral features of the input signal are extracted and are compared with a database of previously examined signals, to which perceptually classified equalization changes have been made. A nearest neighbour optimization is used to map the similarity in spectral features to relevant equalization settings. In [9], a method is presented which automatically sets the release and threshold of a noise gate applied to drum recordings. This work is expanded here. This is an autoadaptive effect. The distortion to the target signal and the residual noise are extracted from the input signal. An objective function is defined which is a weighted combination of these two features. The objective function is minimised subject to weighting parameter, mapping the features to the release and threshold.

Automatic audio effects for musical applications generally have a user input which takes subjective considerations into account. For example, [5] has a global panning width control and [6] has a maximum attenuation control. The panning values output by the automatic panner are scaled between the center, and the user-defined global panning width. The maximum attenuation control defines the maximum gain reduction that can be applied to channels in order to reduce masking with the target channel. If the use of an audio effect cannot be defined in a purely objective way, it is advisable to decouple subjective and objective elements when attempting to automate it. In the case of a noise gate this distinction can be made clearly. The objective is to reduce the amount of noise, so the gate should attenuate the signal when noise is prevalent and should not attenuate when the wanted signal is prevalent. The subjective element is the level of attenuation that should be applied.

2. Method

2.1. Noise Gates in Drum Recordings. A noise gate has five main parameters: threshold (T), attack (A), release (R), hold (H), and gain (G). Threshold and gain are measured in decibels, and attack, release, and hold are measured in seconds. The threshold is the level above which the signal will open the gate and below which it will not. The gain is the attenuation applied to the signal when the gate is closed. The attack is a time constant representing the speed at which the gate opens. The release is a time constant representing the speed at which the gate closes. The hold parameter defines the minimum time for which the gate must remain open. It prevents the gate from switching between states too quickly which can cause modulation artifacts.

A typical drum kit comprises kick drum, snare, hi-hats, cymbals, and any number of tom toms. An example microphone setup will include a kick drum microphone, a snare microphone (possibly two), a microphone for each tom tom, and a set of stereo-overheads to capture a natural mix of the entire kit. In some instances a hi-hat microphone will also be used. When mixing the recording, the overheads will be used as a starting point. The signals from the other microphones are mixed into this to provide emphasis on the main rhythmic components, that is, the kick, snare, and tom toms. Processing is applied to these signals to obtain the desired sound. Compression is invariably used on kick drum recordings. A compressor raises the level of low amplitude regions in the signal, relative to high amplitude regions which has the affect of amplifying the bleed. Noise gates are used to reduce (or remove) bleed from the signal before processing is applied.

Figure 1(a) shows an example kick drum recording containing bleed from secondary sources. Figure 1(b) shows the amplitude envelope of the kick drum contained within the recording, and Figures 1(c) and 1(d) show the amplitude envelope of bleed contained within the signal. The large and small spikes up to 1.875 seconds in Figure 1(c) are snare hits and the final two large spikes are tom-tom hits. Figure 1(d) has reduced limits on the y-axis. This figure shows the cymbal hit at 0 seconds, and hi-hat hits, for example, at 1.625 seconds. The amplitude of these parts of the bleed is very low and will have minimal affect on the gate settings. Components of the bleed signal which coincide with the kick drum cannot be removed by the gate (because it is opened by the kick drum). The snare hits coincide with the decay phase of the kick drum hits and so will have the biggest impact on the noise gate time constants. If the release time is short, the gate will be tightly closed before the snare hit, but the natural decay of the kick drum will be choked.

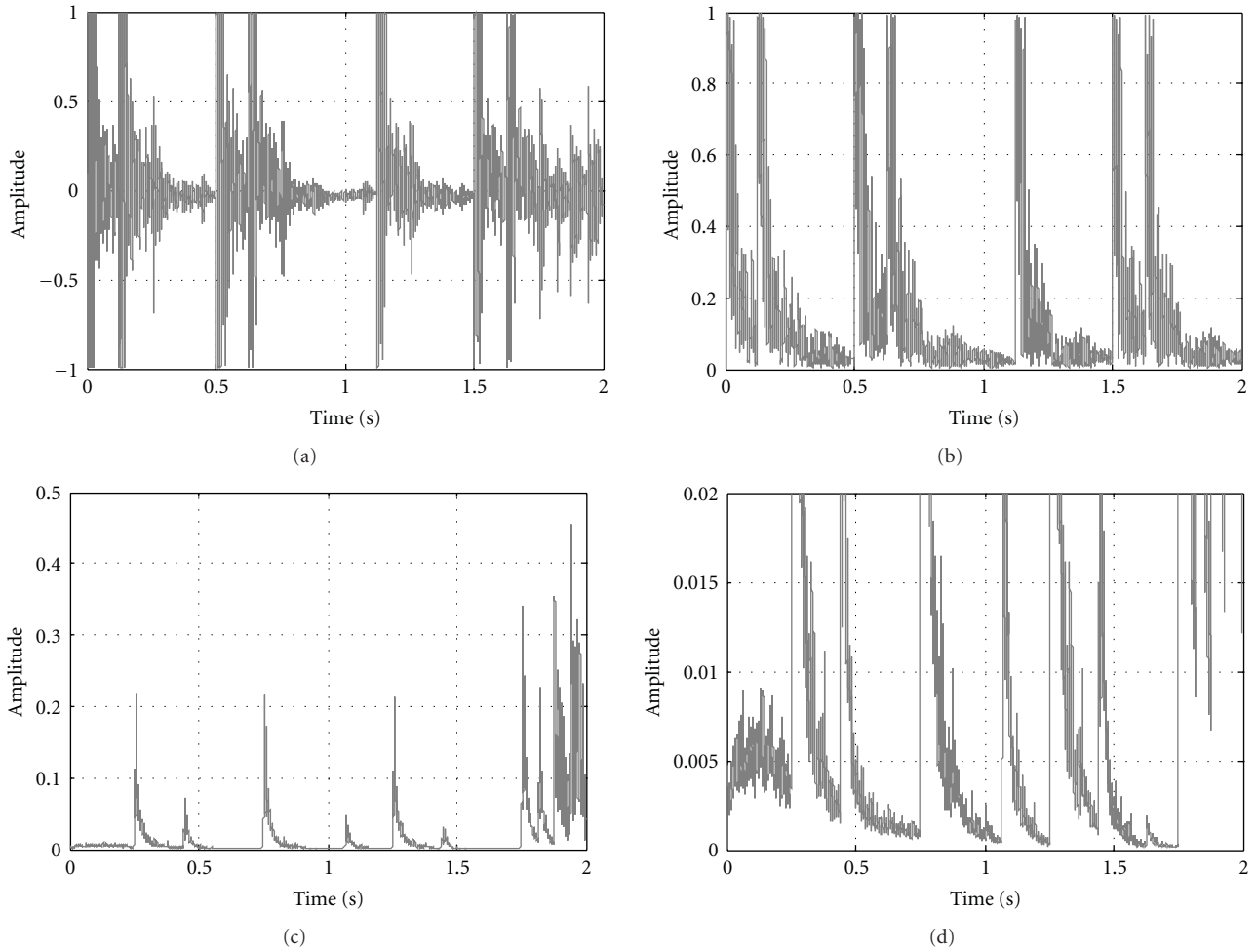


FIGURE 1: An example kick drum recording, (a) is a noisy microphone signal which includes kick drum and bleed, (b) shows the amplitude envelope of the kick drum contained within the noisy signal, and (c) and (d) show the amplitude envelope of the bleed contained within the noisy signal. Part (d) has reduced limits on the y-axis to show cymbals and hi-hats in the bleed signal.

If the release time is long the gate will remain partially open, and the snare hit will be audible to some extent, but the kick drum hit will be allowed to decay more naturally. If the threshold is below the peak amplitude of any part of the bleed signal, then the bleed will open the gate and will be audible. It is necessary to strike a balance between reducing the level of bleed and minimising distortion of the kick drum.

2.2. Audio Files, Artifacts, and Noise Reduction. Audio files representatives of a kick drum recording containing bleed from hi-hats, snare drum, cymbal, and tom toms are investigated. The audio is generated using the commercial software BFD2 from FXpansion. In this software the samples for each drum have been recorded with all microphones active so natural bleed is available. Test audio files are made by soloing the output of the kick drum microphone. Audio files are sequenced by the author. The kick drum signal which contains bleed is referred to as the noisy signal, $\mathbf{y}_n[n]$. This is a combination of the clean kick drum signal $\mathbf{y}_k[n]$ and the bleed signal $\mathbf{y}_b[n]$,

$$\mathbf{y}_n[n] = \mathbf{y}_k[n] + \mathbf{y}_b[n], \tag{1}$$

where $[n]$ is the sample index. $[n]$ will be dropped from this point onward for clarity. Time domain vectors are identified by lowercase, bold, typeface. Passing a signal through the noise gate will generate a gate function, $\mathbf{g}$. This vector contains the gain to be applied to each sample of the input signal. An example gate function is plotted in Figure 1(a). The gate function will generate distortion artifacts in the kick drum signal, D_A,

$$D_A = \frac{\left|\left|(1-\mathbf{g})^T .* \mathbf{y}_k\right|\right|^2}{||\mathbf{y}_k||^2}, \tag{2}$$

and will reduce the bleed signal to a residual level, D_B,

$$D_B = \frac{||\mathbf{g}^T .* \mathbf{y}_b||^2}{||\mathbf{y}_b||^2}, \tag{3}$$

where $.*$ is the elementwise, vector multiplication operator. The signal to artifact ratio (SAR) and the reduction in the bleed level (δ_{bleed}) are given by

$$\begin{aligned} \text{SAR} &= 20\log_{10}(D_A^{-1}), \\ \delta_{\text{bleed}} &= 20\log_{10}(D_B). \end{aligned} \tag{4}$$

In [9] it is proposed that optimal noise gate settings should be found by minimising an objective function which is a weighted combination of the distortion artifacts D_A and the noise reduction D_N. The weighting parameter is then used to control the strength of the gate. The release and threshold are parameters in the objective function, but attack, gain, and hold are fixed. The attack is set to the minimum time of 1 ms, the gain to $-\infty$ dB, and the hold to a value that prevents distortion. A usable automatic gate requires these parameters to be included, in particular the gain setting, which if fixed at $-\infty$ dB will choke the kick drum sound severely. The implementation presented in this paper also includes the attack time and hold time as parameters in the objective function. The gain is used in place of the weighting parameter to control the strength of the gate. Rather than minimising an objective function which contains the distortion artifacts and the residual noise, the distortion artifacts are minimised (SAR is maximised), subject to the reduction in the bleed being greater than some threshold.

2.3. Approximating Distortion Artifacts and Noise Reduction. The distortion artifacts and noise reduction cannot be evaluated without separating the kick and bleed components of the signal. The human auditory system can do this instinctively. A human user will have prior knowledge of what the clean signal sounds like, that is, the user will know that the clean signal is a kick drum. This is replicated when automating the noise gate by inputting a single, clean, kick drum hit to the algorithm. In practice this could be obtained during a sound check, or could be taken from a database of kick drum samples.

The noisy signal is split into windows of quaver length. Each window is attributed to kick or bleed. The divisions within the noisy signal are made based on note onsets. Onsets are identified manually, but it is assumed that they could be identified exactly using an onset detection algorithm. The work in [10] is a benchmark paper on onset detection, and [11] contains a summary of drum transcription and source separation techniques. The spectral power of each window of the noisy signal is correlated with the spectral power of a region of the clean kick drum signal of equal length. If the correlation is above a predefined threshold, it is attributed to kick drum. The correlation is calculated as the scalar product of the normalised spectral powers. $\mathbf{X}_i$ is the spectral power of window i of the noisy signal, and $\mathbf{X}_c$ is the spectral power of the clean kick drum signal. The correlation is given by

$$\mathbf{c}_i = \left(\frac{\mathbf{X}_i}{\|\mathbf{X}_i\|}\right)^T \cdot \left(\frac{\mathbf{X}_c}{\|\mathbf{X}_c\|}\right), \tag{5}$$

where $\mathbf{c}_i$ is the correlation of the spectral powers of window i of the noisy signal with the clean kick drum signal. Windows of the noisy signal with a correlation greater than the threshold of 0.95 are assigned to kick drum. All other windows are assigned to bleed. An approximation of the clean signal is made by aligning a copy of the clean kick drum hit with the start of each window assigned to kick drum. This forms the synthesized clean signal $\mathbf{y}_z$, which is used in place of $\mathbf{y}_k$ in (2). The bleed is approximated by silencing all windows in the noisy signal which are attributed to the kick drum.

Figure 2 shows how the approximations to the kick and bleed components in the noisy signal are obtained. Figure 2(a) shows the noisy signal. It has been quantized with an eighth note quantization grid and windows are based on this spacing. Figure 2(d) of this figure shows the correlations between the spectral power of each window in the noisy signal with the spectral power of the clean kick drum hit. Marked on this figure is the correlation threshold of 0.95. All windows which contain a kick drum hit have a correlation above this threshold. Figures 2(b) and 2(c) show the synthesized kick drum signal, $\mathbf{y}_z$, and the approximate bleed signal, $\mathbf{y}_b$, respectively. The dotted lines on Figures 2(a) and 2(c) show the gate function $\mathbf{g}$, which is the gain applied by the gate as the noisy signal passes through it. The dotted line on Figure 1(b) shows the function $(1-\mathbf{g})$. These are used to estimate the distortion artifacts and the residual noise as defined in (2) and (3).

2.4. The Noise Gate Optimization Algorithm. Common practice when using a noise gate to reduce bleed in drum tracks is to first set the gain to $-\infty$ dB. The threshold is then set as low as possible to allow the maximum amount of kick drum to pass through without allowing the gate to be opened by the bleed signal. The release is set as slow as possible whilst ensuring that the gate is closed before the onset of any bleed notes. For very fast tempos this may not be possible without introducing significant artifacts, in which case some bleed notes which occur close to the kick drum hit may be allowed to pass through. The implications of this in the automatic implementation will be discussed later. It is assumed that the gate must be closed for all bleed onsets. The attack is set to the fastest value which does not introduce any distortion artifacts. The hold time is continually adjusted to remove modulation artifacts caused by rapid opening and closing of the gate. During an interonset interval assigned to kick drum, the gate should go through one attack phase and one release phase only. The hold parameter should be as low as possible whilst maintaining this requirement. If it is too long it can affect the release phase of the gate. Once all other parameters have been set, the gain is adjusted subjectively to the desired level.

Figure 3 is a flowchart of the algorithm. The inputs on the left are constraints enforced at each stage. The inputs on the right are the parameter values at each stage. The signal is split into regions which contain kick drum and regions which contain bleed, as discussed in Section 2.3. An initial estimate of the threshold is found by maximising the SAR, subject to the constraint that the bleed level is reduced by at least 60 dB. This is identified by the parameter

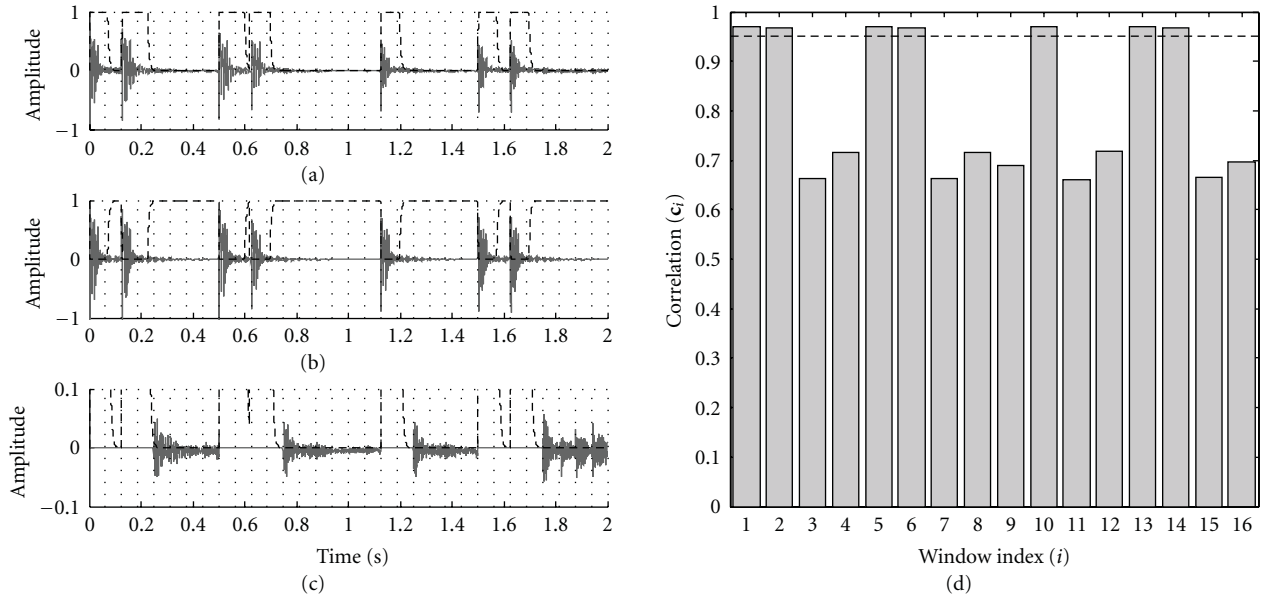


FIGURE 2: Approximations to the kick drum and bleed signals, (a) contains the noisy signal $\mathbf{y}_n$, (b) contains the synthesized clean kick drum signal $\mathbf{y}_z$, (c) contains the component of the signal attributed to bleed $\mathbf{y}_b$, and (d) shows the correlation of the spectral power of each window with the spectral power of the clean kick drum signal. The correlation threshold is identified by the dotted line.

δ_{bleed}, which is the minimum change in the bleed level after gating. The attack, release, and hold are set to their minimum values during the initial threshold estimate and the gain is set for full signal attenuation ($G = 0$ on a linear scale). This ensures that the threshold is set to the lowest feasible value. The minimum hold time is found which permits only one attack phase and one release phase for each kick drum window. These constraints are identified by parameters N_{attack} and N_{release} which correspond to the permitted number of attack and release phases, respectively. The other gate inputs are the minimum values of attack and release and the initial threshold estimate. The threshold estimate is required because the minimum hold time can vary significantly with threshold. The threshold is then recalculated using the updated hold parameter. Finally the attack and release are found by maximising the SAR, subject to the bleed reduction. Steepest descent gradient methods are used to minimise functions at each stage.

Breaking the algorithm into stages rather than defining a single objective function which contains all parameters has a significant advantage in this kind of optimization scheme. The major problems when using a single objective function are discontinuous regions in the solution space and regions of the solution space which have zero sensitivity with respect to small changes to the parameters. This is the case for all parameters when the threshold is close to zero (at which point the signal level is always above the threshold). By optimising each parameter in turn, and ensuring that the start point lies within a sensitive, continuous region at each stage, this problem is overcome. Alternative optimization methods which do not rely on gradient information could potentially be used.

3. Results

The algorithm is tested using a simple drum beat. The tempo of the beat is 120 bpm, the time signature is 4/4, and the kick hits lie on a 1/8 note quantization grid. There are some 1/16 note snare drum hits, but none of these occur immediately after a kick drum hit. This ensures that each kick drum window has a length of 1/8 note. The required bleed reduction is set to $\delta_{\text{bleed}} = -60$ dB, and the gain of the noise gate is set to $-\infty$ dB, that is, full attenuation. Figures 4(a) and 4(b) show the signal before and after gating, respectively. The gate function is plotted with a dashed line. It can be seen that the kick drum decay phase of the gated kick drum has been shortened, so that the signal level is approximately zero at the beginning of the region assigned to bleed, which occurs at 0.5 s. A user would now be free to adjust the gain parameter with the automated threshold, attack, release, and hold to change the strength of the gate.

The automatic noise gate algorithm is now investigated for a range of required bleed reductions, and for a range of noisy signals which contained different strengths of bleed. The strength of the bleed is measured relative to the test case described above, and includes bleed strengths of +0 dB, +2 dB, +4 dB, and +6 dB. Figures 5(a)–5(d) contain plots of the threshold, release, hold, and SAR, respectively. The attack has not been plotted because in all cases the algorithm set it to the minimum value of 1 ms.

Initial discussions are focused on the signal with a relative bleed strength of +0 dB. Figure 5(a) shows that the threshold has a stepped profile, and that it decreases as the required bleed reduction is decreased. Table 1 shows the peak levels extracted from each region of the noisy signal attributed to bleed. The overall peak level is −28 dB, which occurs in

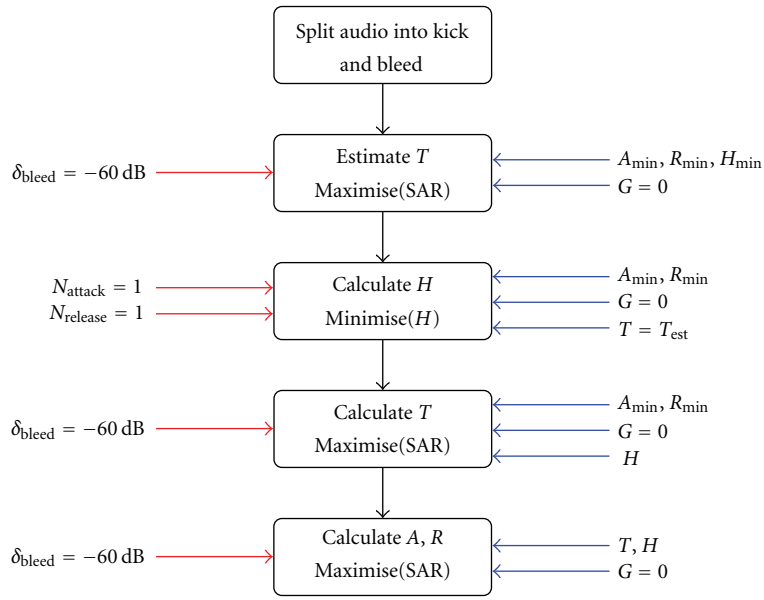


Figure 3: Automatic noise gate flow chart.

the final section and is due to the tom tom hits. Inspection of Figure 5(a) shows that the threshold is above this for $\delta_{\text{bleed}} < -10$ dB, and so the bleed signal will not open the gate. Large reductions in bleed, for example, $\delta_{\text{bleed}} = -60$ dB, result in thresholds which are higher than the peak level of the bleed by around 3 dB. This headroom is required to ensure that the gate has sufficient time to close during the release phase (which in calculating the threshold is set to the minimum value of 10 ms). As the required reduction in bleed becomes smaller, the gate does not need to be closed so tightly by the end of the release phase, which permits a lower threshold. The threshold follows a stepped profile because the bleed reduction is highly sensitive to small changes in the threshold. The threshold is set using the predetermined hold time and minimum attack and release times, as shown in Figure 3. Using these parameter values, a change in the threshold from −25.89 dB, to −22.56 dB results in a change in δ_{bleed} from −22.5 dB to −56.4 dB. With the tolerance used, there are no intermediate threshold values that will give a bleed reduction between −22.5 dB and −56.4 dB. When the strength of the bleed is increased, a similar trend can be seen, but the difference between the threshold and the peak level of the bleed (shown in Table 1) gets progressively smaller. This is because with a higher strength of bleed, the absolute reduction in bleed to produce the same relative change is smaller, and the gate does not need to be closed so tightly by the end of the release phase.

For a fixed threshold the release time gradually increases as the required bleed reduction decreases. This is expected because the gate does not need to be closed so tightly by the start of the bleed window. Each step drop in threshold causes a sudden shortening of the time between the start of the release phase and the start of the following bleed window and so a step drop in release time is needed to produce the required bleed reduction.

Table 1: Peak signal level in the bleed regions identified by t_1 and t_2 for a range of relative bleed strengths.

t_1	t_2	0 dB	+2 dB	+4 dB	+6 dB
0.5	1	−29.1	−26.3	−25.6	−24.9
1.5	2.25	−29.1	−28.7	−28.2	−27.6
2.5	3	−29.3	−28.9	−28.4	−29.7
3.5	4	−28.0	−26.5	−24.5	−22.7

The hold time gives what appears to be the most unintuitive results. For signals with relative bleed strengths of +0 dB, +2 dB, and +4 dB, the hold time remains roughly constant at around 40 ms. The signal which has a bleed strength of +6 dB has a far lower hold time when the required bleed reduction is large, and shows a sudden increase in hold time when $\delta_{\text{bleed}} > -20$ dB. The value of the hold time will depend on the degree to which the envelope of the kick drum signal is fluctuating about the threshold. If there are substantial fluctuations a longer hold time is required. The hold time is determined using the initial estimate of the threshold. Signals with different relative bleed strengths have different initial threshold estimates. Evidently for the signal with a bleed strength of +6 dB, there are minimal fluctuations in the envelope of the kick drum signal about the initial threshold estimate when the required bleed reduction is large. When the required bleed reduction is decreased, the initial threshold estimate is lower, and there are more fluctuations in the envelope of the kick drum signal about it. A longer hold time is therefore needed.

The SAR generally increases as the required reduction in bleed decreases. This is expected. A gentler gate causes less distortion in to kick drum signal. There are a few anomalous points where a decrease in the required bleed reduction is accompanied by an decrease in the SAR.

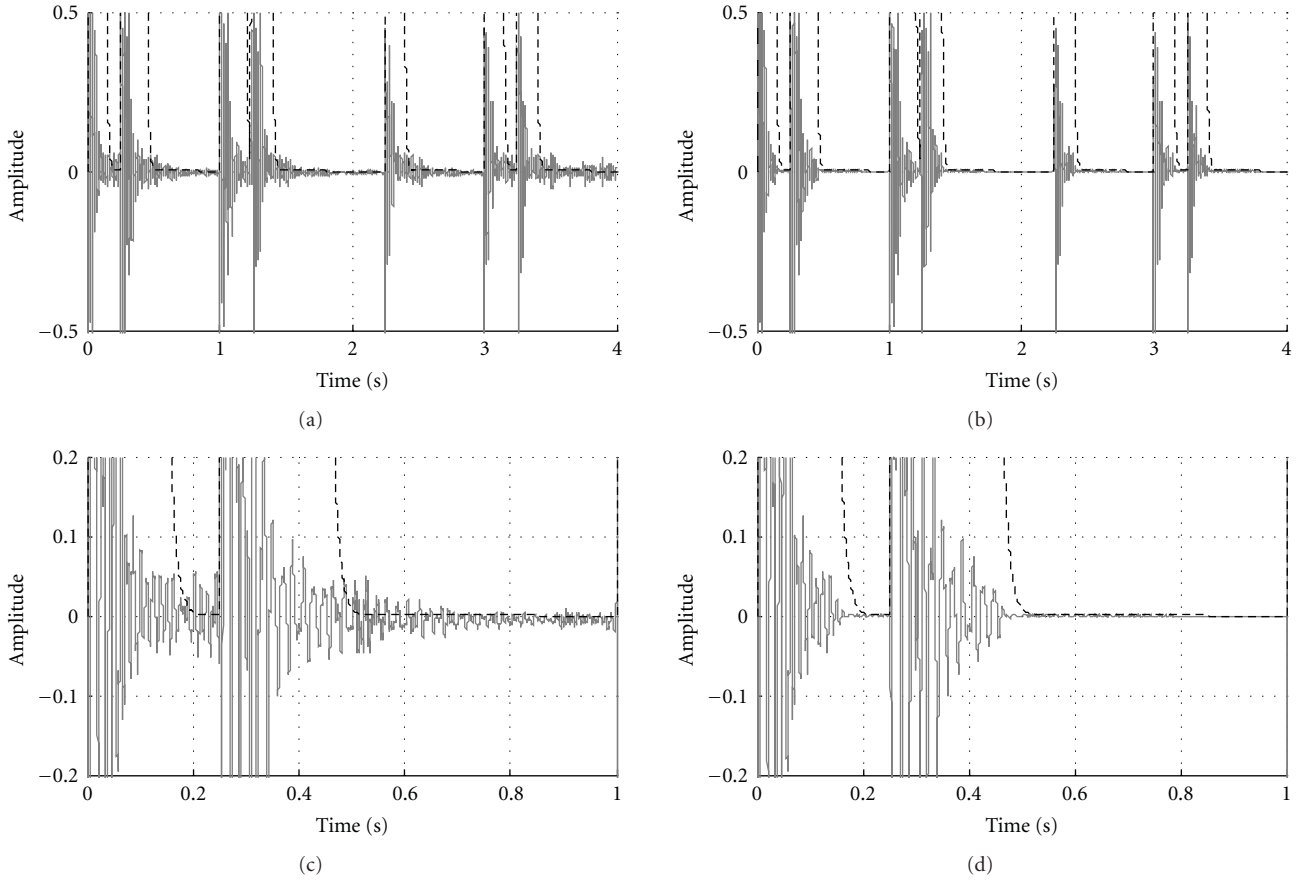


Figure 4: Kick drum recording before and after gating, (a) before gating, and (b) after gating, with $\delta_{\text{bleed}} = -60$ dB.

These points coincide with step reductions in the threshold and release. It is suggested that in these transitional points a smoother change in the release and threshold may be required. This cannot be achieved with the algorithm in its current form because the threshold and release time are evaluated independently. It may be possible to include an additional, final stage which optimizes all of the parameters together.

4. Discussion

In designing the algorithm, manual use of a noise gate has been taken into account. It is the opinion of the author that by replicating the human thought process, the automated results should better approximate those obtained by a human user. Although formal evaluation has not been undertaken, informal testing has shown this to be the case.

The algorithm has been designed so that it is independent of the specific noise gate implementation. It would be easier to develop an algorithm if hidden aspects of the implementation, such as the transient filter properties, and the level detector, were known, but this would limit the use of the algorithm to a specific noise gate. This approach also ties in with the concept of replicating human operation because the parameters are set based only on the input and output of the gate and so much like with a human user, decisions are based purely on changes to the properties of the signal. It is the opinion of the author that this black box approach has most potential when considering commercial developments in the automation of any audio effect, as it allows the automation algorithm to be developed independently of the effect implementation (so long as the same parameters are available).

The algorithm presented divides the signal into a number of intervals based on the position of onsets. Problems will arise with drum recordings at high tempos and with high resolution quantization grids. In these cases it is likely that the kick drum regions will be very short, resulting in a choked kick drum sound after gating. A human operator would adjust the release to allow some bleed onsets which are close to the kick drum hit to pass through. This should be incorporated into the automatic gating algorithm. This could be done by defining a minimum kick drum window length, based on the amplitude envelope of the clean kick drum hit.

It is interesting to consider how the automatic noise gate presented in this paper fits into the A-DAFx framework. Most A-DAFx have a small analysis frame and update control parameters continuously, more or less in real time. This is particularly the case with established auto-adapative effects such as compressors. The algorithm presented here uses an audio segment of around 8 seconds, and takes 5–10 seconds to form and minimise the objective function. Despite this

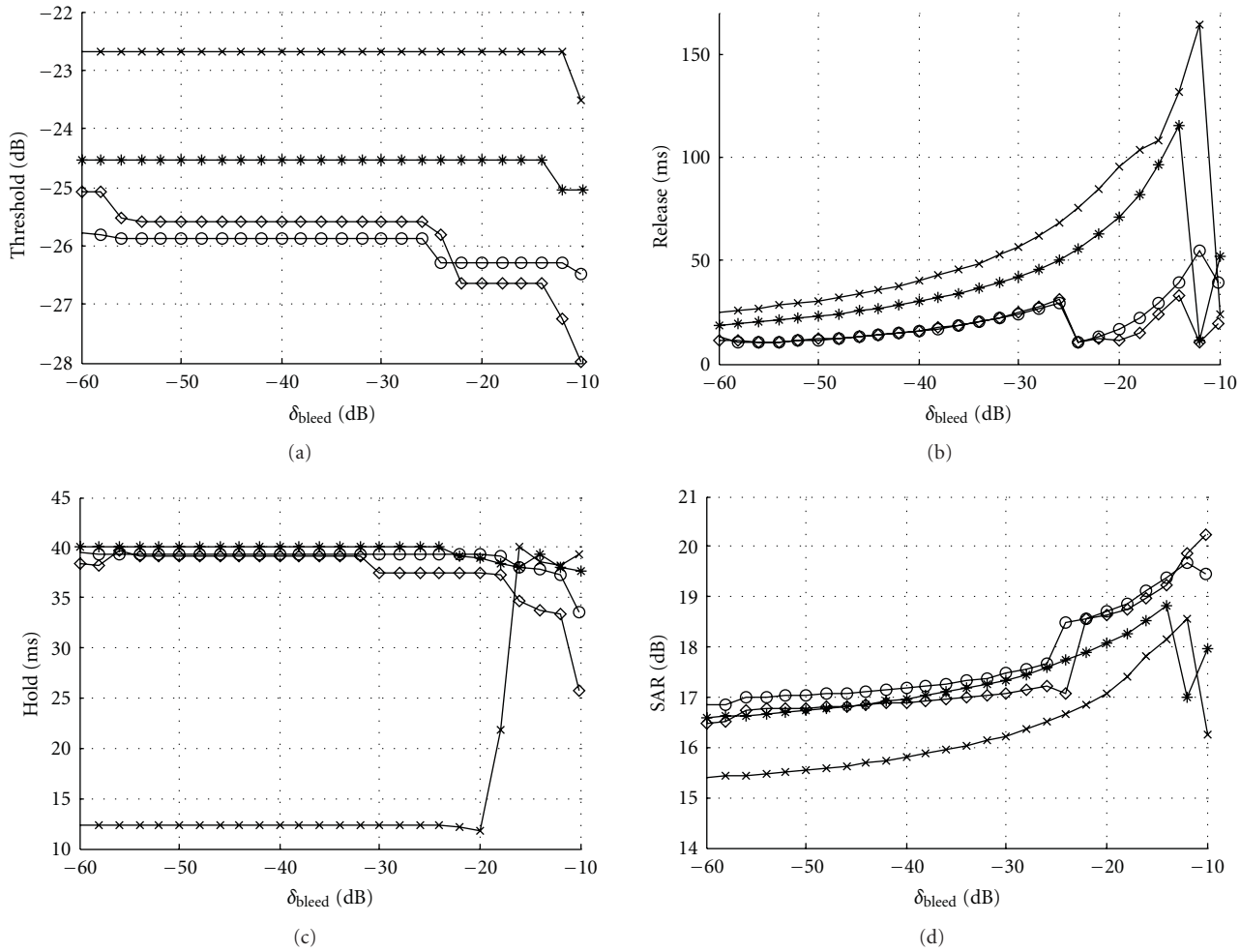


Figure 5: Noise gate parameter values after optimization, plotted against the required reduction in bleed (δ_{bleed} as defined in Figure 3). Part (a) shows threshold, (b) shows release time, (c) shows hold time and (d) shows SAR. Results are plotted for a number of relative bleed strengths identified by, ◇: +0 dB, ◯: +2 dB, ∗: +4 dB, ×: +6 dB.

lengthy time frame the algorithm could still be implemented within the A-DAFx framework. Large and sudden changes to noise gate parameters are undesirable, so an accumulative learning approach could be used as in [7].

Subjective evaluation has not yet been performed for this work. It would be useful to compare the values of the gate parameters output by the algorithm to those of an experienced engineer. This could be used to determine suitable reductions in SNR to be used in the algorithm, which may or may not be based on properties of the input signal.

5. Conclusions

An algorithm has been presented which automatically sets the threshold, release, attack, and hold parameters of a noise gate used on a kick drum recording that contains bleed from secondary sources. The parameters identified cause minimal distortion to the kick signal, whilst enforcing a predefined reduction in the level of the bleed signal. The gain parameter is not set automatically and is used to manually control the strength of the gate. The algorithm has been developed independently from the noise gate implementation, and through consideration of the process followed by a human user. It has been tested for signals with varying levels of bleed, and varying amounts of bleed reduction. The gate settings found are intuitively correct, although as yet no subjective evaluation has been undertaken to compare them to expert users.

Acknowledgment

The authors would like to thank the EPSRC for funding this research.

References

[1] U. Zolzer, *Digital Audio Effects*, John Wiley & Sons, New York, NY, USA, 2002.

[2] V. Verfaille, U. Zolzer, and D. Arfib, "Adaptive digital audio effects (A-DAFx): a new class of sound transformations," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1817–1831, 2006.

[3] D. Dugan, "Automatic microphone mixing," in *Proceedings of the AES 51st International Convention*, 1975.

[4] S. Julstrom and T. Tichy, "Direction-sensitive gating: a new approach to automatic mixing," in *Proceedings of the AES 73rd International Convention*, 1976.

[5] E. Perez-Gonzalez and J. Reiss, "Automatic mixing: live down-mixing stereo panner," in *Proceedings of the 10th International Conference on Digital Audio Effects (DAFX '07)*, 2007.

[6] E. Perez-Gonzalez and J. Reiss, "Improved control for selective minimization of masking using inter-channel dependancy effects," in *Proceedings of the 11th International Conference on Digital Audio Effects (DAFX '08)*, 2008.

[7] E. Perez-Gonzalez and J. Reiss, "Automatic gain and fader control for live mixing," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '09)*, pp. 1–4, October 2009.

[8] D. Reed, "Perceptual assistant to do sound equalization," in *Proceedings of the International Conference on Intelligent User Interfaces (IUI '00)*, pp. 212–218, January 2000.

[9] M. Terrell and J. Reiss, "Automatic noise gate settings for multitrack drum recordings," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFX '09)*, September 2009.

[10] A. Klapuri, "Sound onset detection by appluing psychoacoustic knowledge," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '99)*, pp. 115–118, Phoenix, Ariz, USA, 1999.

[11] D. FitzGerald, *Automatic drum transcription and source separation*, Ph.D. thesis, Dublin Institute of Technology, 2004.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 712749, 10 pages
doi:10.1155/2010/712749

Research Article

High-Quality Time Stretch and Pitch Shift Effects for Speech and Audio Using the Instantaneous Harmonic Analysis

Elias Azarov,[1] Alexander Petrovsky (EURASIP Member),[1,2] and Marek Parfieniuk (EURASIP Member)[2]

[1] *Department of Computer Engineering, Belarussian State University of Informatics and Radioelectronics, 220050 Minsk, Belarus*
[2] *Department of Real-Time Systems, 15-351 Bialystok University of Technology, Bialystok, Poland*

Correspondence should be addressed to Alexander Petrovsky, palex@bsuir.by

Received 6 May 2010; Accepted 10 November 2010

Academic Editor: Udo Zoelzer

The paper presents methods for instantaneous harmonic analysis with application to high-quality pitch, timbre, and time-scale modifications. The analysis technique is based on narrow-band filtering using special analysis filters with frequency-modulated impulse response. The main advantages of the technique are high accuracy of harmonic parameters estimation and adequate harmonic/noise separation that allow implementing audio and speech effects with low level of audible artifacts. Time stretch and pitch shift effects are considered as primary application in the paper.

1. Introduction

Parametric representation of audio and speech signals has become integral part of modern effect technologies. The choice of an appropriate parametric model significantly defines overall quality of implemented effects. The present paper describes an approach to parametric signal processing based on deterministic/stochastic decomposition. The signal is considered as a sum of periodic (harmonic) and residual (noise) parts. The periodic part can be efficiently described as a sum of sinusoids with slowly varying amplitudes and frequencies, and the residual part is assumed to be irregular noise signal. This representation was introduced in [1] and since then has been profoundly studied and significantly enhanced. The model provides good parameterization of both voiced and unvoiced frames and allows using different modification techniques for them. It insures effective and simple processing in frequency domain; however, the crucial point there is accuracy of harmonic analysis. The harmonic part of the signal is specified by sets of harmonic parameters (amplitude, frequency, and phase) for every instant of time. A number of methods have been proposed to estimate these parameters. The majority of analysis methods assume local stationarity of amplitude and frequency parameters within the analysis frame [2, 3]. It makes the analysis procedure easier but, on the other hand, degrades parameters estimation and periodic/residual separation accuracy.

Some good alternatives are methods that make estimation of instantaneous harmonic parameters. The notion of instantaneous frequency was introduced in [4, 5], the estimation methods have been presented in [4–9]. The aim of the current investigation is to study applicability of the instantaneous harmonic analysis technique described in [8, 9] to a processing system for making audio and speech effects (such as pitch, timbre, and time-scale modifications). The analysis method is based on narrow-band filtering using analysis filters with closed form impulse response. It has been shown [8] that analysis filters can be adjusted in accordance with pitch contour in order to get adequate estimate of high-order harmonics with rapid frequency modulations. The technique presented in this paper has the following improvements:

(i) simplified closed form expressions for instantaneous parameters estimation;
(ii) pitch detection and smooth pitch contour estimation;
(iii) improved harmonic parameters estimation accuracy.

The analysed signal is separated into periodic and residual parts and then processed through modification techniques. Then the processed signal can be easily synthesized

in time domain at the output of the system. The deterministic/stochastic representation significantly simplifies the processing stage. As it is shown in the experimental section, the combination of the proposed analysis, processing, and synthesis techniques provides good quality of signal analysis, modification, and reconstruction.

2. Time-Frequency Representations and Harmonic Analysis

The sinusoidal model assumes that the signal $s(n)$ can be expressed as the sum of its periodic and stochastic parts:

$$s(n) = \sum_{k=1}^{K} \mathrm{MAG}_k(n) \cos \varphi_k(n) + r(n), \tag{1}$$

where $\mathrm{MAG}_k(n)$—the instantaneous magnitude of the kth sinusoidal component, K is the number of components, $\varphi_k(n)$ is the instantaneous phase of the kth component, and $r(n)$ is the stochastic part of the signal. Instantaneous phase $\varphi_k(n)$ and instantaneous frequency $f_k(n)$ are related as follows:

$$\varphi_k(n) = \sum_{i=0}^{n} \frac{2\pi f_k(i)}{F_s} + \varphi_k(0), \tag{2}$$

where F_s is the sampling frequency and $\varphi_k(0)$ is the initial phase of the kth component. The harmonic model states that frequencies $f_k(n)$ are integer multiples of the fundamental frequency $f_0(n)$ and can be calculated as

$$f_k(n) = k f_0(n). \tag{3}$$

The harmonic model is often used in speech coding since it represents voiced speech in a highly efficient way. The parameters $\mathrm{MAG}_k(n)$, $f_k(n)$, and $\varphi_k(0)$ are estimated by means of the sinusoidal (harmonic) analysis. The stochastic part obviously can be calculated as the difference between the source signal and estimated sinusoidal part:

$$r(n) = s(n) - \sum_{k=1}^{K} \mathrm{MAG}_k(n) \cos \varphi_k(n). \tag{4}$$

Assuming that sinusoidal components are stationary (i.e., have constant amplitude and frequency) over a short period of time that correspond to the length of the analysis frame, they can be estimated using DFT:

$$S(f) = \frac{1}{N} \sum_{n=0}^{N-1} s(n) e^{-j2\pi nf/N}, \tag{5}$$

where N is the length of the frame. The transformation gives spectral representation of the signal by sinusoidal components of multiple frequencies. The balance between frequency and time resolution is defined by the length of the analysis frame N. Because of the local stationarity assumption DFT can hardly provide accurate estimate of frequency-modulated components that gives rise to such approaches as harmonic transform [10] and fan-chirp transform [11]. The general idea of these approaches is using the Fourier transform of the warped-time signal.

The signal warping can be carried out before transformation or directly embedded in the transform expression [11]:

$$S(\omega, \alpha) = \sum_{n=-\infty}^{\infty} s(n) \sqrt{|1 + \alpha n|} e^{-j\omega(1+(1/2)\alpha n)n}, \tag{6}$$

where ω is frequency and α is the chirp rate. The transform is able to identify components with linear frequency change; however, their spectral amplitudes are assumed to be constant. There are several methods for estimation instantaneous harmonic parameters. Some of them are connected with the notion of analytic signal based on the Hilbert transform (HT). A unique complex signal $z(t)$ from a real one $s(t)$ can be generated using the Fourier transform [12]. This also can be done as the following time-domain procedure:

$$z(t) = s(t) + jH[s(t)] = a(t) e^{j\varphi(t)}, \tag{7}$$

where H is the Hilbert transform, defined as

$$H[s(t)] = \mathrm{p.v.} \int_{-\infty}^{+\infty} \frac{s(t-\tau)}{\pi\tau} d\tau, \tag{8}$$

where p.v. denotes Cauchy principle value of the integral. $z(t)$ is referred to as Gabor's complex signal, and $a(t)$ and $\varphi(t)$ can be considered as the instantaneous amplitude and instantaneous phase, respectively. Signals $s(t)$ and $H[s(t)]$ are theoretically in quadrature. Being a complex signal $z(t)$ can be expressed in polar coordinates, and therefore $a(t)$ and $\varphi(t)$ can be calculated as follows:

$$\begin{aligned} a(t) &= \sqrt{s^2(t) + H^2[s(t)]}, \\ \varphi(t) &= \arctan\left(\frac{H[s(t)]}{s(t)}\right). \end{aligned} \tag{9}$$

Recently the discrete energy separation algorithm (DESA) based on the Teager energy operator was presented [5]. The energy operator is defined as

$$\Psi[s(n)] = s^2(n) - s(n-1)s(n+1), \tag{10}$$

where the derivative operation is approximated by the symmetric difference. The instantaneous amplitude $\mathrm{MAG}(n)$ and frequency $f(n)$ can be evaluated as

$$\begin{aligned} \mathrm{MAG}(n) &= \frac{2\Psi[s(n)]}{\sqrt{\Psi[s(n+1) - s(n-1)]}}, \\ f(n) &= \arcsin \sqrt{\frac{\Psi[s(n+1) - s(n-1)]}{4\Psi[s(n)]}}. \end{aligned} \tag{11}$$

The Hilbert transform and DESA can be applied only to monocomponent signals as long as for multicomponent signals the notion of a single-valued instantaneous frequency and amplitude becomes meaningless. Therefore, the signal should be split into single components before using these techniques. It is possible to use narrow-band filtering for this purpose [6]. However, in the case of frequency-modulated components, it is not always possible due to their wide frequency.

3. Instantaneous Harmonic Analysis

3.1. Instantaneous Harmonic Analysis of Nonstationary Harmonic Components. The proposed analysis method is based on the filtering technique that provides direct parameters estimation [8]. In voiced speech harmonic components are spaced in frequency domain and each component can be limited thereby a narrow frequency band. Therefore harmonic components can be separated within the analysis frame by filters with nonoverlapping bandwidths. These considerations point to the applicability and effectiveness of the filtering approach to harmonic analysis. The signal $s(n)$ is represented as a sum of bandlimited cosine functions with instantaneous amplitude, phase, and frequency. It is assumed that harmonic components are spaced in frequency domain so that each component can be limited by a narrow frequency band. The harmonic components can be separated within the analysis frame by filters with nonoverlapping bandwidths. Let us denote the number of cosines L and frequency separation borders (in Hz) $F_0 \le F_2 \le \cdots \le F_L$, where $F_0 = 0$, $F_L = F_s/2$. The given signal $s(n)$ can be represented as its convolution with the impulse response of the ideal low-pass filter $h(n)$:

$$\begin{aligned} s(n) &= s(n) * h(n) = s(n) * \frac{\sin(\pi n)}{n\pi} \\ &= s(n) * \int_{-0.5}^{0.5} \cos(2\pi f n)\, df \\ &= s(n) * \left[2 \int_{0}^{0.5} \cos(2\pi f n)\, df \right] \\ &= s(n) * \left[\sum_{k=1}^{L} \frac{2}{F_s} \int_{F_{k-1}}^{F_k} \cos\left(2\pi f \frac{n}{F_s}\right) df \right] \\ &= \sum_{k=1}^{L} s(n) * \left[\frac{2}{F_s} h_k(n) \right] = \sum_{k=1}^{L} s_k(n), \end{aligned} \tag{12}$$

where $h_k(n)$—the impulse response of the band-pass filter with passband $[F_{k-1}, F_k]$, $s_k(n)$—bandlimited output signal. The impulse response can be written in the following way:

$$\begin{aligned} h_k(n) &= \int_{F_{k-1}}^{F_k} \cos\left(2\pi f \frac{n}{F_s}\right) df \\ &= \begin{cases} 2F_\Delta^k, & n = 0, \\ \dfrac{F_s}{n\pi} \cos\left(\dfrac{2\pi n}{F_s} F_c^k\right) \sin\left(\dfrac{2\pi n}{F_s} F_\Delta^k\right), & n \ne 0, \end{cases} \end{aligned} \tag{13}$$

where $F_c^k = (F_{k-1} + F_k)/2$ and $F_\Delta^k = (F_k - F_{k-1})/2$. Parameters F_c^k and F_Δ^k correspond to the center frequency of the passband and the half of bandwidth, respectively. Convolution of finite signal $s(n)$ $(0 \le n \le N-1)$ and $h_k(n)$ can be expressed as the following sum:

$$s_k(n) = \sum_{i=0}^{N-1} \frac{2s(i)}{\pi(n-i)} \cos\left(\frac{2\pi(n-i)}{F_s} F_c^k\right) \sin\left(\frac{2\pi(n-i)}{F_s} F_\Delta^k\right). \tag{14}$$

The expression can be rewritten as a sum of zero frequency components:

$$s_k(n) = A(n)\cos(0n) + B(n)\sin(0n), \tag{15}$$

where

$$\begin{aligned} A(n) &= \sum_{i=0}^{N-1} \frac{2s(i)}{\pi(n-i)} \sin\left(\frac{2\pi(n-i)}{F_s} F_\Delta^k\right) \cos\left(\frac{2\pi(n-i)}{F_s} F_c^k\right), \\ B(n) &= \sum_{i=0}^{N-1} \frac{-2s(i)}{\pi(n-i)} \sin\left(\frac{2\pi(n-i)}{F_s} F_\Delta^k\right) \sin\left(\frac{2\pi(n-i)}{F_s} F_c^k\right). \end{aligned} \tag{16}$$

Thus, considering (15), the expression (14) is a magnitude and frequency-modulated cosine function:

$$s_k(n) = \mathrm{MAG}(n)\cos(\varphi(n)), \tag{17}$$

with instantaneous magnitude $\mathrm{MAG}(n)$, phase $\varphi(n)$, and frequency $f(n)$ that can be calculated as

$$\begin{aligned} \mathrm{MAG}(n) &= \sqrt{A^2(n) + B^2(n)}, \\ \varphi(n) &= \arctan\left(\frac{-B(n)}{A(n)}\right), \\ f(n) &= \frac{\varphi(n+1) - \varphi(n)}{2\pi} F_s. \end{aligned} \tag{18}$$

In that way the signal frame $s(n)$ $(0 \le n \le N-1)$ can be represented by L cosines with instantaneous amplitude and frequency. Instantaneous sinusoidal parameters of the filter output are available at every instant of time within the analysis frame. The filter output $s_k(n)$ can be interpreted as an analytical signal $s_k^a(n)$ in the following way:

$$s_k^a(n) = A(n) + jB(n). \tag{19}$$

The bandwidth specified by border frequencies F_{k-1} and F_k (or by parameters F_c^k and F_Δ^k) should cover the frequency of the periodic component that is being analyzed. In many applications there is no need to represent entire signal as a sum of modulated cosines. In hybrid parametric representation it is necessary to choose harmonic components with smooth contours of frequency and amplitude values. For accurate sinusoidal parameters estimation of periodical components with high-frequency modulations a frequency-modulated filter can be used. The closed form impulse response of the filter is modulated according to frequency contour of the analyzed component. This approach is quite applicable to analysis of voiced speech since rough harmonic frequency trajectories can be estimated from the pitch contour. Considering centre frequency of the filter bandwidth as a function of time $F_c(n)$, (15) can be rewritten in the following form:

$$s_k(n) = A(n)\cos(0n) + B(n)\sin(0n), \tag{20}$$

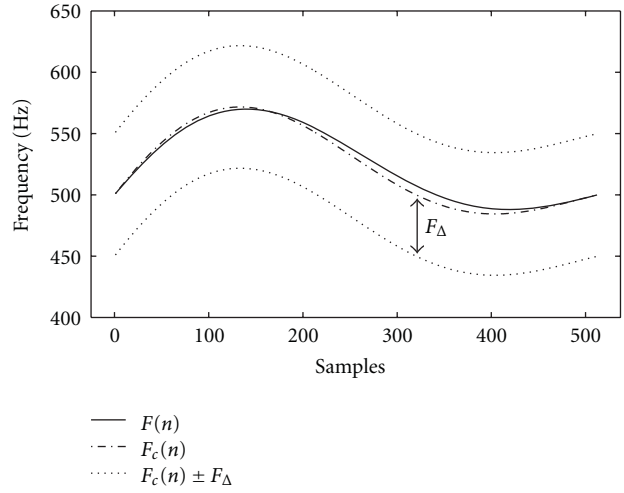


Figure 1: Frequency-modulated analysis filter $N = 512$.

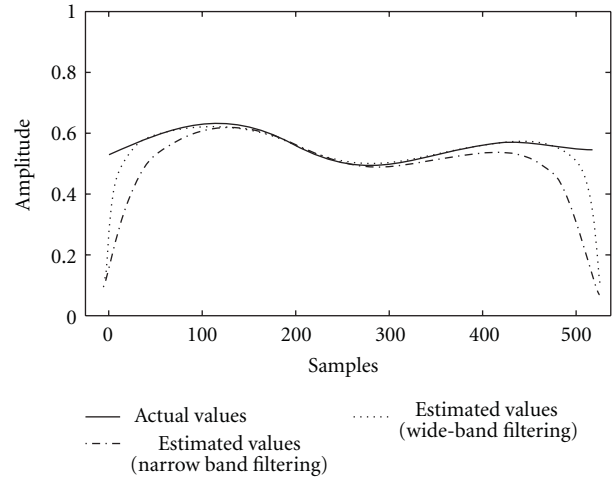


Figure 2: Instantaneous amplitude estimation accuracy.

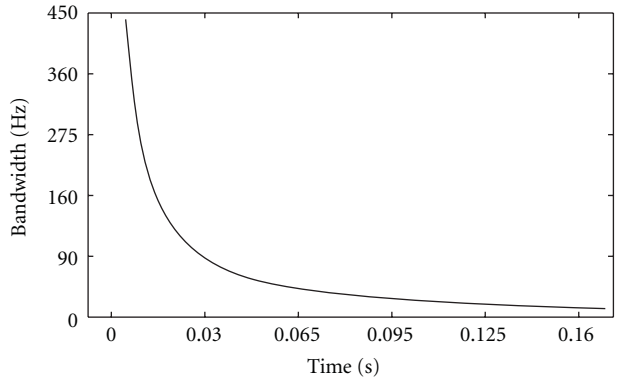


Figure 3: Minimal bandwidth of analysis filter.

where

$$A(n) = \sum_{i=0}^{N-1} \frac{2s(i)}{\pi(n-i)} \sin\left(\frac{2\pi(n-i)}{F_s} F_\Delta^k\right) \cos\left(\frac{2\pi}{F_s}\varphi_c(n,i)\right),$$

$$B(n) = \sum_{i=0}^{N-1} \frac{-2s(i)}{\pi(n-i)} \sin\left(\frac{2\pi(n-i)}{F_s} F_\Delta^k\right) \sin\left(\frac{2\pi}{F_s}\varphi_c(n,i)\right),$$

$$\varphi_c(n,i) = \begin{cases} \sum_{j=n}^{i} F_c^k(j), & n < i, \\ -\sum_{j=i}^{n} F_c^k(j), & n > i, \\ 0, & n = i. \end{cases} \tag{21}$$

The required instantaneous parameters can be calculated using expressions (18). The frequency-modulated filter has a warped band pass, aligned to the given frequency contour $F_c^k(n)$ that provides adequate analysis of periodic components with rapid frequency alterations. This approach is an alternative to time warping that is used in speech analysis [11]. In Figure 1 an example of parameters estimation is shown. The frequency contour of the harmonic component can be covered by the filter band pass specified by the centre frequency contour $F_c^k(n)$ and the bandwidth $2F_\Delta^k$.

Center frequency contour $F_c(n)$ is adjusted within the analysis frame providing narrow-band filtering of frequency-modulated components.

3.2. Filter Properties. Estimation accuracy degrades close to borders of the frame because of signal discontinuity and spectral leakage. However, the estimation error can be reduced using wider passband—Figure 2.

In any case the passband should be wide enough in order to provide adequate estimation of harmonic amplitudes. If the passband is too narrow, the evaluated amplitude values become lower than they are in reality. It is possible to determine the filter bandwidth as a threshold value that gives desired level of accuracy. The threshold value depends on length of analysis window and type of window function. In Figure 3 the dependence for Hamming window is presented, assuming that amplitude attenuation should be less than -20 dB.

It is evident that required bandwidth becomes more narrow when the length of the window increases. It is also clear that a wide passband affects estimation accuracy when the signal contains noise. The noise sensitivity of the filters with different bandwidths is demonstrated in Figure 4.

3.3. Estimation Technique. In this subsection the general technique of sinusoidal parameters estimation is presented. The technique does not assume harmonic structure of the signal and therefore can be applied both to speech and audio signals [13].

In order to locate sinusoidal components in frequency domain, the estimation procedure uses iterative adjustments of the filter bands with a predefined number of iterations—Figure 5. At every step the centre frequency of each filter is changed in accordance with the calculated frequency value in order to position energy peak at the centre of the band. At

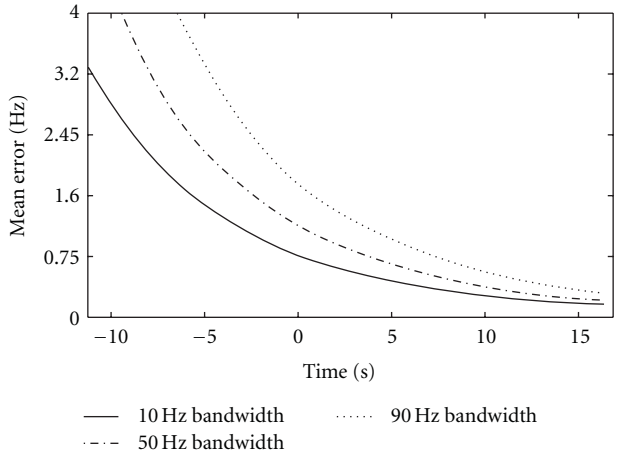


Figure 4: Instantaneous frequency estimation error.

the initial stage, the frequency range of the signal is covered by overlapping bands $B_1, \ldots, B_h$ (where h is the number of bands) with constant central frequencies $F_C^{B_1}, \ldots, F_C^{B_h}$, respectively. At every step the respective instantaneous frequencies $f^{B_1}(n_c), \ldots, f^{B_h}(n_c)$ are estimated by formulas (15) and (18) at the instant that corresponds to the centre of the frame n_c. Then the central bandwidth frequencies are reset $F_C^{B_x} = f^{B_x}(n_c)$, and the next estimation is carried out. When all the energy peaks are located, the final sinusoidal parameters (amplitude, frequency, and phase) can be calculated using the expressions (15) and (18) as well. During the peak location process, some of the filter bands may locate the same component. Duplicated parameters are discarded by comparison of the centre band frequencies $F_C^{B_1}, \ldots, F_C^{B_h}$.

In order to discard short-term components (that apparently are transients or noise and should be taken to the residual), sinusoidal parameters are tracked from frame to frame. The frequency and amplitude values of adjacent frames are compared, providing long-term component matching. The technique has been used in the hybrid audio coder [13], since it is able to pick out the sinusoidal part and leave the original transients in the residual without any prior transient detection. In Figure 6 a result of the signal separation is presented. The source signal is a jazz tune (Figure 6(a)).

The analysis was carried out using the following settings: analysis frame length—48 ms, analysis step—14 ms, filter bandwidths—70 Hz, and windowing function—the Hamming window. The synthesized periodic part is shown in Figure 6(b). As can be seen from the spectrogram, the periodic part contains only long sinusoidal components with high-energy localization. The transients are left untouched in the residual signal that is presented in Figure 6(c).

3.4. Speech Analysis. In speech processing, it is assumed that signal frames can be either voiced or unvoiced. In voiced segments the periodical constituent prevails over the noise, in unvoiced segments the opposite takes place, and therefore any harmonic analysis is unsuitable in that case. In the proposed analysis framework voiced/unvoiced frame classification is carried out using pitch detector. The harmonic parameters estimation procedure consists of the two following stages:

(i) initial fundamental frequency contour estimation;

(ii) harmonic parameters estimation with fundamental frequency adjustment.

In voiced speech analysis, the problem of initial fundamental frequency estimation comes to finding a periodical component with the lowest possible frequency and sufficiently high energy. Within the possible fundamental frequency range (in this paper, it is defined as [60, 1000] Hz) all periodical components are extracted, and then the suitable one is considered as the fundamental. In order to reduce computational complexity, the source signal is filtered by a low-pass filter before the estimation.

Having fundamental contour estimated, it is possible to calculate filter impulse responses aligned to the fundamental frequency contour. Central frequency of the filter band is calculated as the instantaneous frequency of fundamental multiplied by the number k of the correspondent harmonic $F_C^k(n) = kf_0(n)$. The procedure goes from the first harmonic to the last, adjusting fundamental frequency at every step—Figure 7. The fundamental frequency recalculation formula can be written as follows:

$$f_0(n) = \sum_{i=0}^{k} \frac{f_i(n)\mathrm{MAG}_i(n)}{(i+1)\sum_{j=0}^{k}\mathrm{MAG}_j(n)}. \tag{22}$$

The fundamental frequency values become more precise while moving up the frequency range. It allows making proper analysis of high-order harmonics with significant frequency modulations. Harmonic parameters are estimated using expressions (10)-(11). After parameters estimation, the periodical part of the signal is synthesized by formula (1) and subtracted from the source in order to get the noise part.

In order to test applicability of the proposed technique, a set of synthetic signals with predefined parameters was used. The signals were synthesized with different harmonic-to-noise ratio defined as

$$\mathrm{HNR} = 10\lg\frac{\sigma_H^2}{\sigma_e^2}, \tag{23}$$

where σ_H^2 is the energy of the deterministic part of the signal and σ_e^2 is the energy of its stochastic part. All the signals were generated using a specified fundamental frequency contour $f_0(n)$ and the same number of harmonics—20. Stochastic parts of the signals were generated as white noise with such energy that provides specified HNR values. After analysis the signals were separated into stochastic and deterministic parts with new harmonic-to-noise ratios:

$$\widetilde{\mathrm{HNR}} = 10\lg\frac{\widetilde{\sigma}_H^2}{\sigma_e^2}. \tag{24}$$

Quantitative characteristics of accuracy were calculated as signal-to-noise ratio:

$$\mathrm{SNR}_H = 10\lg\frac{\widetilde{\sigma}_H^2}{\sigma_{eH}^2}, \tag{25}$$

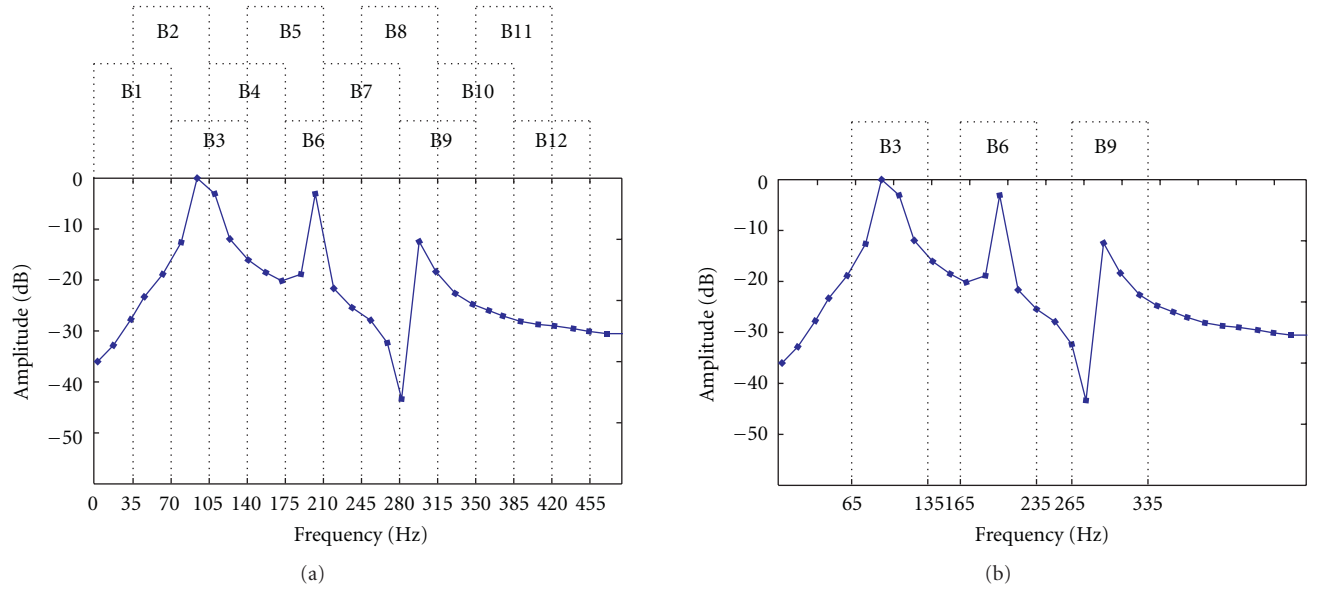


Figure 5: Sinusoidal parameters estimation using analysis filters: (a) initial frequency partition; (b) frequency partition after second iteration.

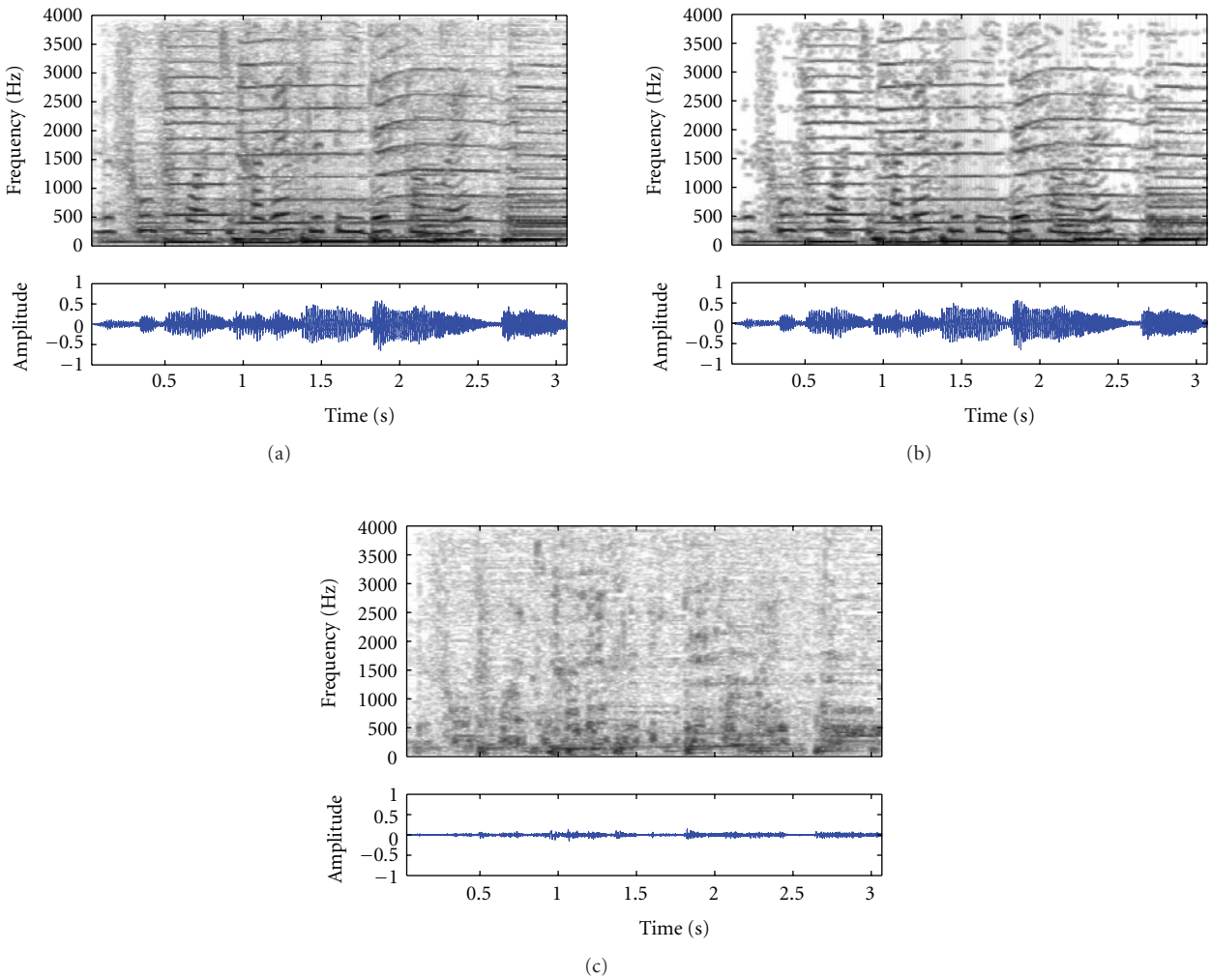


Figure 6: Periodic/stochastic separation of an audio signal: (a) source signal; (b) periodic part; (c) stochastic part.

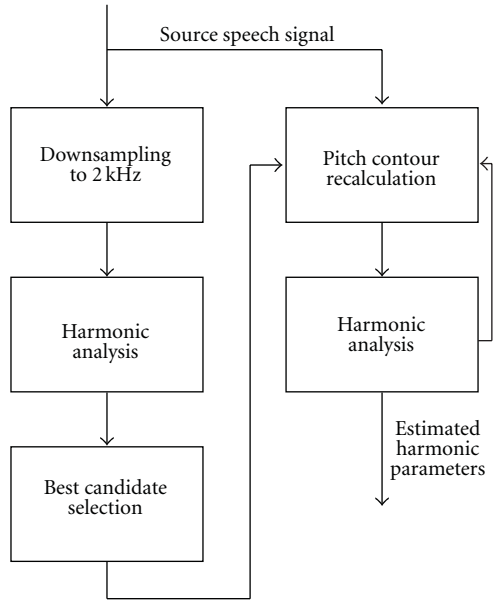


Figure 7: Harmonic analysis of speech.

where $\tilde{\sigma}_H^2$—energy of the estimated harmonic part and σ_{eH}^2—energy of the estimation error (energy of the difference between source and estimated harmonic parts). The signals were analyzed using the proposed technique and STFT-based harmonic transform method [10]. During analysis the same frame length was used (64 ms) and the same window function (Hamming window). In both methods, it was assumed that the fundamental frequency contour is known and that frequency trajectories of the harmonics are integer multiples of the fundamental frequency. The results, reported in Table 1 show that the measured SNR_H values decrease with HNR values. However, for nonstationary signals, the proposed technique provides higher SNR_H values even when HNR is low.

An example of natural speech analysis is presented in Figure 8. The source signal is a phrase uttered by a female speaker ($F_s = 8$ kHz). Estimated harmonic parameters were used for the synthesis of the signal's periodic part that was subtracted from the source in order to get the residual. All harmonics of the source are modeled by the harmonic analysis when the residual contains transient and noise components, as can be seen in the respective spectrograms.

4. Effects Implementation

The harmonic analysis described in the previous section results in a set of harmonic parameters and residual signal. Instantaneous spectral envelopes can be estimated from the instantaneous harmonic amplitudes and the fundamental frequency obtained at the analysis stage [14]. The linear interpolation can be used for this purpose. The set of frequency envelopes can be considered as a function $E(n, f)$ of two parameters: sample number and frequency. Pitch shifting procedure affects only the periodic part of the signal that can be synthesized as follows:

$$s(n) = \sum_{k=1}^{K} E\left(n, \overline{f}_k(n)\right) \cos \overline{\varphi}_k(n). \tag{26}$$

Phases of harmonic components $\overline{\varphi}_k(n)$ are calculated according to a new fundamental frequency contour $\overline{f}_0(n)$:

$$\overline{\varphi}_k(n) = \sum_{i=0}^{n} \frac{2\pi \overline{f}_k(i)}{F_s} + \overline{\varphi}_k^{\Delta}(n). \tag{27}$$

Harmonic frequencies are calculated by formula (3):

$$\overline{f}_k(n) = k\overline{f}_0(n). \tag{28}$$

Additional phase parameter $\overline{\varphi}_k^{\Delta}(n)$ is used in order to keep the original phases of harmonics relative phase of the fundamental

$$\overline{\varphi}_k^{\Delta}(n) = \varphi_k(n) - k\varphi_0(n). \tag{29}$$

As long as described pitch shifting does not change spectral envelope of the source signal and keeps relative phases of the harmonic components, the processed signal has a natural sound with completely new intonation. The timbre of speakers voice is defined by the spectral envelope function $E(n, f)$. If we consider the envelope function as a matrix

$$E = \begin{pmatrix} E(0,0) & \cdots & E\left(0, \frac{F_s}{2}\right) \\ \vdots & \ddots & \vdots \\ E(N,0) & \cdots & E\left(N, \frac{F_s}{2}\right) \end{pmatrix}, \tag{30}$$

then any timbre modification can be expressed as a conversion function $C(E)$ that transforms the source envelope matrix E into a new matrix $\overline{E}$:

$$\overline{E} = C(E). \tag{31}$$

Since the periodic part of the signal is expressed by harmonic parameters, it is easy to synthesize the periodic part slowing down or stepping up the tempo. Amplitude and frequency contours should be interpolated in the respective moments of time, and then the output signal can be synthesized. The noise part is parameterized by spectral envelopes and then time-scaled as described in [15]. Separate periodic/noise processing provides high-quality time-scale modifications with low level of audible artifacts.

5. Experimental Results

In this section an example of vocal processing is shown. The concerned processing system is aimed at pitch shifting in order to assist a singer.

The voice of the singer is analyzed by the proposed technique and then synthesized with pitch modifications to assist the singer to be in tune with the accompaniment. The target pitch contour is predefined by analysis of a reference

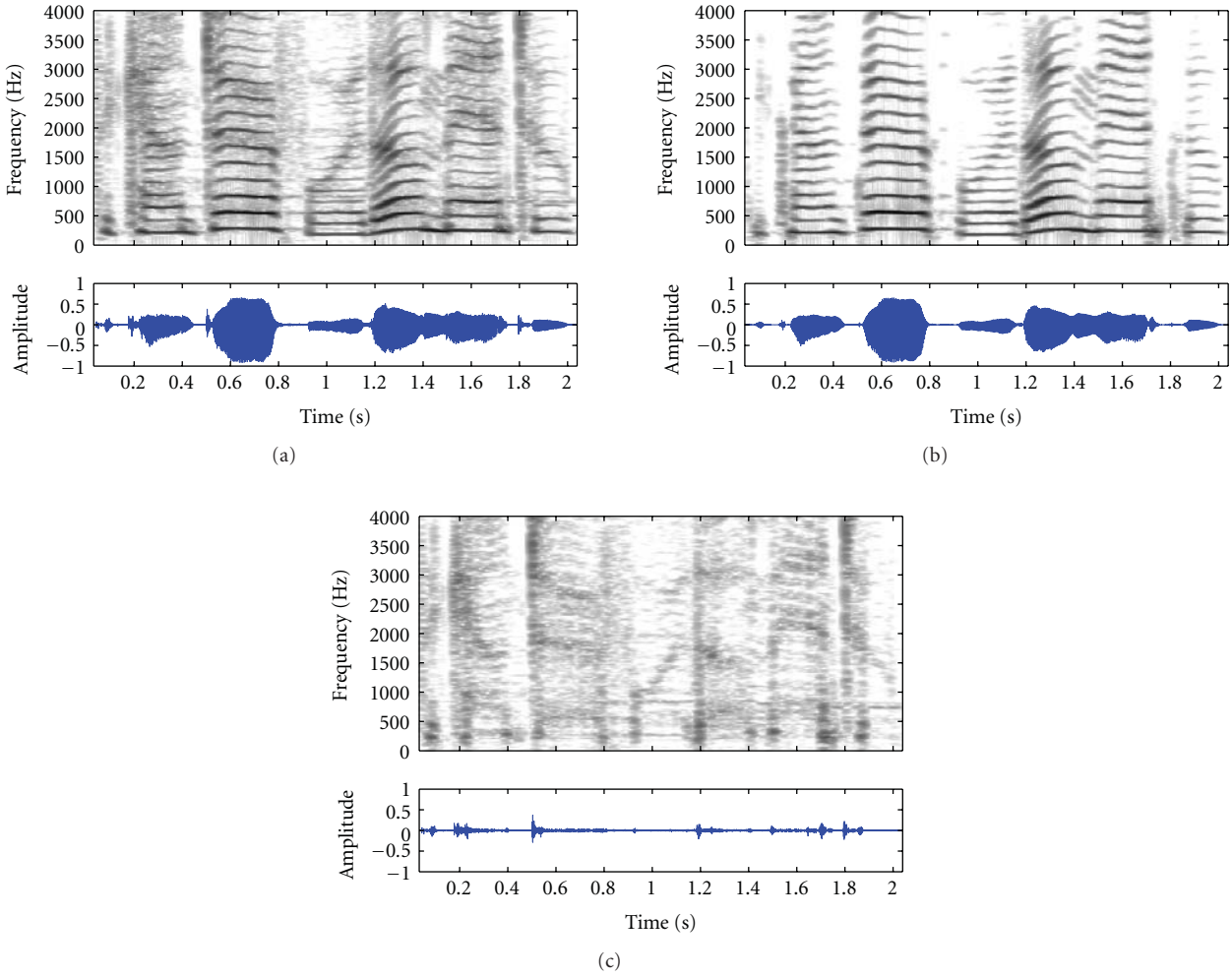


Figure 8: Periodic/stochastic separation of an audio signal: (a) source signal; (b) periodic part; (c) stochastic part.

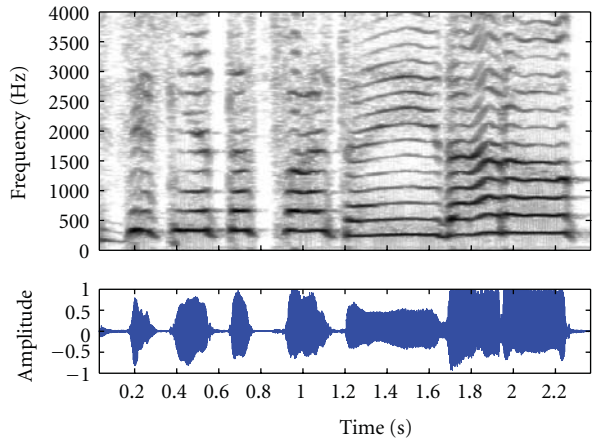


Figure 9: Reference signal.

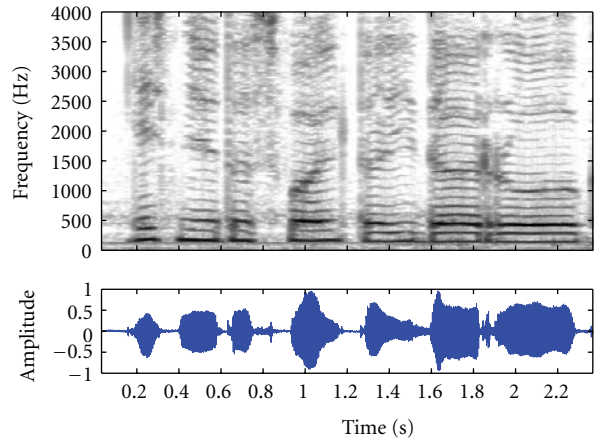


Figure 10: Source signal.

recording. Since only pitch contour is changed, the source voice maintains its identity. The output signal however is damped in regions, where the energy of the reference signal is low in order to provide proper synchronization with accompaniment. The reference signal is shown in Figure 9, it is a recorded male vocal. The recording was made in a studio

Table 1: Results of synthetic speech analysis.

	Harmonic transform method		Instantaneous harmonic analysis	
HNR	$\widetilde{\text{HNR}}$	SNR_H	$\widetilde{\text{HNR}}$	SNR_H
Signal 1—$f_0(n)$ = 150 Hz for all n, random constant harmonic amplitudes				
∞	41.5	41.5	50.4	50.4
40	38.5	41.4	41.2	44.7
20	20.8	29.2	21.9	26.2
10	10.7	19.5	11.9	16.4
0	1.2	9.2	2.9	6.0
Signal 2—$f_0(n)$ changes from 150 to 220 Hz at a rate of 0.1 Hz/ms, constant harmonic amplitudes that model sound [a]				
∞	41.5	41.5	48.3	48.3
40	38.2	40.7	41.0	44.3
20	21.0	29.5	22.1	26.4
10	11.0	20.3	12	17.1
0	1.3	9.3	2.7	6.5
Signal 3—$f_0(n)$ changes from 150 to 220 Hz at a rate of 0.1 Hz/ms, variable harmonic amplitudes that model sequence of vowels				
∞	19.6	19.7	34.0	34.0
40	17.3	17.5	31.2	31.8
20	17.7	21.3	20.1	25.5
10	8.7	15.6	10.3	15.1
0	−0.8	7.55	0.94	5.2
Signal 4—$f_0(n)$ changes from 150 to 220 Hz at a rate of 0.1 Hz/ms, variable harmonic amplitudes that model sequence of vowels, harmonic frequencies deviate from integer multiplies of $f_0(n)$ on 10 Hz				
∞	13.2	14.0	26.9	27.0
40	10.6	11.9	24.8	25.3
20	11.9	13.6	19.3	22.7
10	6.9	12.1	9.6	14
0	−1.6	6.1	0.5	4.2

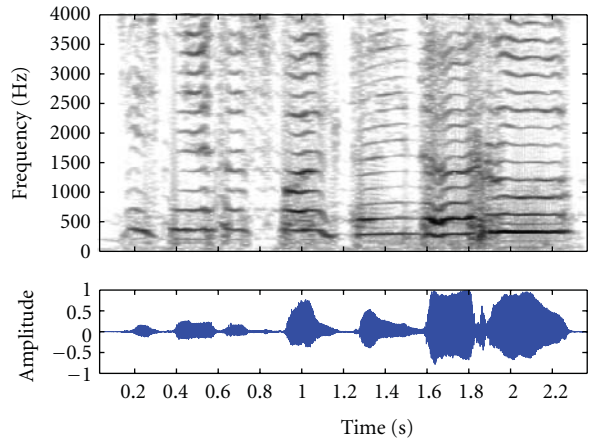


Figure 11: Output signal.

with a low level of background noise. The fundamental frequency contour was estimated from the reference signal as described in Section 3. As can be seen from Figure 10, the source vocal has different pitch and is not completely noise free.

The source signal was analyzed using proposed harmonic analysis, and then the pitch shifting technique was applied as has been described above.

The synthesized signal with pitch modifications is shown in Figure 11. As can be seen the output signal contains the pitch contour of the reference signal, but still has timbre, and energy of the source voice. The noise part of the source signal (including background noise) remained intact.

6. Conclusions

The stochastic/deterministic model can be applied to voice processing systems. It provides efficient signal parameterization in the way that is quite convenient for making voice effects such as pitch shifting, timbre and time-scale modifications. The practical application of the proposed harmonic analysis technique has shown encouraging results. The described approach might be a promising solution

to harmonic parameters estimation in speech and audio processing systems [13].

Acknowledgment

This work was supported by the Polish Ministry of Science and Higher Education (MNiSzW) in years 2009–2011 (Grant no. N N516 388836).

References

[1] T. F. Quatieri and R. J. McAulay, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1449–1464, 1986.

[2] A. S. Spanias, "Speech coding: a tutorial review," *Proceedings of the IEEE*, vol. 82, no. 10, pp. 1541–1582, 1994.

[3] X. Serra, "Musical sound modeling with sinusoids plus noise," in *Musical Signal Processing*, C. Roads, S. Pope, A. Pi-cialli, and G. De Poli, Eds., pp. 91–122, Swets & Zeitlinger, 1997.

[4] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal," *Proceedings of the IEEE*, vol. 80, no. 4, pp. 520–568, 1992.

[5] P. Maragos, J. F. Kaiser, and T. F. Quatieri, "Energy separation in signal modulations with application to speech analysis," *IEEE Transactions on Signal Processing*, vol. 41, no. 10, pp. 3024–3051, 1993.

[6] T. Abe, T. Kobayashi, and S. Imai, "Harmonics tracking and pitch extraction based on instantaneous frequency," in *Proceedings of the 20th International Conference on Acoustics, Speech, and Signal Processing*, pp. 756–759, May 1995.

[7] T. Abe and M. Honda, "Sinusoidal model based on instantaneous frequency attractors," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1292–1300, 2006.

[8] E. Azarov, A. Petrovsky, and M. Parfieniuk, "Estimation of the instantaneous harmonic parameters of speech," in *Proceedings of the 16th European Signal Processing Conference (EUSIPCO '08)*, Lausanne, Switzerland, 2008.

[9] I. Azarov and A. Petrovsky, "Harmonic analysis of speech," *Speech Technology*, no. 1, pp. 67–77, 2008 (Russian).

[10] F. Zhang, G. Bi, and Y. Q. Chen, "Harmonic transform," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 151, no. 4, pp. 257–263, 2004.

[11] L. Weruaga and M. Képesi, "The fan-chirp transform for non-stationary harmonic signals," *Signal Processing*, vol. 87, no. 6, pp. 1504–1522, 2007.

[12] D. Gabor, "Theory of communication," *Proceedings of the IEE*, vol. 93, no. 3, pp. 429–457, 1946.

[13] A. Petrovsky, E. Azarov, and A. A. Petrovsky, "Harmonic representation and auditory model-based parametric matching and its application in speech/audio analysis," in *Proceedings of the 126th AES Convention*, p. 13, Munich, Germany, 2009, Preprint 7705.

[14] E. Azarov and A. Petrovsky, "Instantaneous harmonic analysis for vocal processing," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx '09)*, Como, Italy, September 2009.

[15] S. Levine and J. Smith, "A sines+transients+noise audio representation for data compression and time/pitch scale modifications," in *Proceedings of the 105th AES Convention*, San Francisco, Calif, USA, September 1998, Preprint 4781.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 436895, 10 pages
doi:10.1155/2010/436895

Research Article

A Real-Time Semiautonomous Audio Panning System for Music Mixing

Enrique Perez_Gonzalez and Joshua D. Reiss

Centre for Digital Music Centre, School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, UK

Correspondence should be addressed to Enrique Perez_Gonzalez, promix_mac@mac.com

Received 26 January 2010; Accepted 23 April 2010

Academic Editor: Udo Zoelzer

A real-time semiautonomous stereo panning system for music mixing has been implemented. The system uses spectral decomposition, constraint rules, and cross-adaptive algorithms to perform real-time placement of sources in a stereo mix. A subjective evaluation test was devised to evaluate its quality against human panning. It was shown that the automatic panning technique performed better than a nonexpert and showed no significant statistical difference to the performance of a professional mixing engineer.

1. Introduction

Stereo panning aims to transform a set of monaural signals into a two-channel signal in a pseudostereo field [1]. Many methods and panning ratios have been proposed, the most common one being the sine-cosine panning law [2, 3]. In stereo panning the ratio at which its power is spread between the left and the right channels determines the position of the source. Over the years the use of panning on music sources has evolved and some common practices can now be identified.

Now that recallable digital systems have become common, it is possible to develop intelligent expert systems capable of aiding the work of the sound engineer. An expert panning system should be capable of creating a satisfactory stereo mix of multichannel audio by using blind signal analysis, without relying on knowledge of original source locations or other visual or contextual aids.

This paper extends the work first presented in [4]. It presents an expert system capable of blindly characterizing multitrack inputs and semiautonomously panning sources with panning results comparable to a human mixing engineer. This was achieved by taking into account technical constraints and common practices for panning, while minimizing human input. Two different approaches are described and subjective evaluation demonstrates that the semi-autonomous panner has equivalent performance to that of a professional mixing engineer.

2. Panning Constraints and Common Practices

In practice, the placement of sound sources is achieved using a combination of creative choices and technical constraints based on human perception of source localization. It is not the purpose of this paper to emulate the more artistic and subjective decisions in source placement. Rather, we seek to embed the common practices and technical constraints into an algorithm which automatically places sound sources. The idea behind developing an expert semi-autonomous panning machine is to use well-established common rules to devise the spatial positioning of a signal.

(1) When the human expert begins to mix, he or she tends to do it from a monaural, all centered position, and gradually moves the pan pots [5]. During this process, all audio signals are running through the mixer at all times. In other words, source placement is performed in realtime based on accumulated knowledge of the sound sources and the resultant

mix, and there is no interruption to the signal path during the panning process.

(2) Panning is not the result of individual channel decisions; it is the result of an interaction between channels. The audio engineer takes into account the content of all channels, and the interaction between them, in order to devise the correct panning position of every individual channel [6].

(3) The sound engineer attempts to maintain balance across the stereo field [7, 8]. This helps maintain the overall energy of the mix evenly split over the stereo speakers and maximizes the dynamic use of the stereo channels.

(4) In order to minimize spectral masking, channels with similar spectral content are placed apart from each other [6, 9]. This results in a mix where individual sources can be clearly distinguished, and this also helps when the listener uses the movement of his or her head to interpret spatial cues.

(5) Hard panning is uncommon [10]. It has been established that panning a ratio of 8 to 12 dBs is more than enough to achieve a full left or full right image [11]. For this reason, the width of the panning positions is restricted.

(6) Low-frequency content should not be panned. There are two main reasons for doing this. First, it ensures that the low-frequency content remains evenly distributed across speakers [12]. This minimizes audible distortions that may occur in the high-power reproduction of low-frequencies. Second, the position of a low frequency source is often psychoacoustically imperceptible. In general, we cannot correctly localize frequencies lower than 200 Hz [13]. It is thought that this is due to the fact that the use of InterTime Difference as a perceptual clue for localization of low frequency sources is highly dependent on room acoustics and loudspeaker placement, and Inter-Level Differences are not a useful perceptual cue at low frequencies since the head only provides significant attenuation of high-frequency sources [14].

(7) High-priority sources tend to be kept towards the centre, while lower priority sources are more likely to be panned [7, 8]. For instance, the vocalist in a modern pop or rock group (often the lead performer) would often not be panned. This relates to the idea of matching a physical stage setup to the relative positions of the sources.

3. Implementation

3.1. Cross-Adaptive Implementation. The automatic panner is implemented as a cross-adaptive effect, where the output of each channel is determined from analysis of all input channels [15]. For applications that require a realtime signal processing, the signal analysis and feature extraction has been implemented using side chain processing, as depicted in Figure 1. The audio signal flow remains real-time while

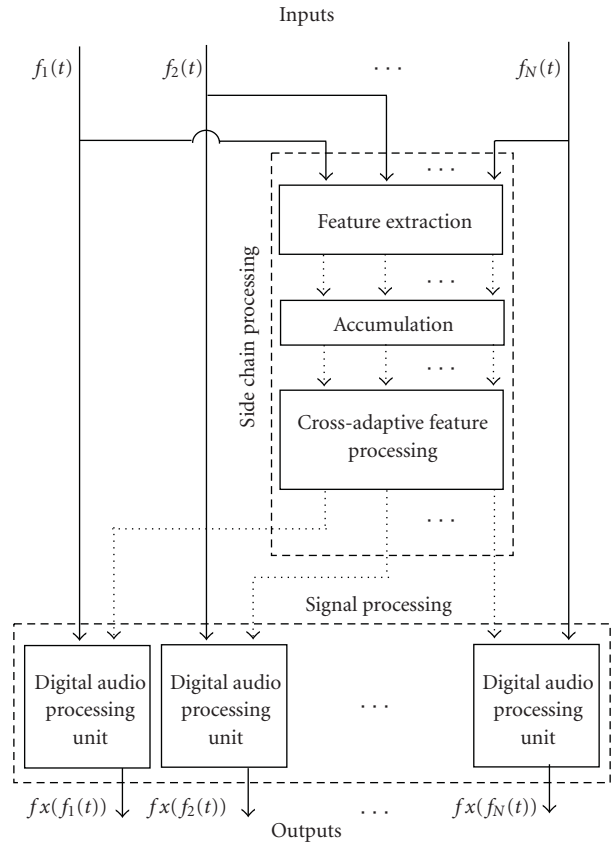


Figure 1: General diagram of a cross-adaptive device using side chain processing with feature accumulation.

the required analysis of the input signals is performed in separate instances. The signal analysis involves accumulating a weighted time average of extracted features. Accumulation allows us to quickly converge on an appropriate panning position in the case of a stationary signal or smoothly adjust the panning position as necessary in the case of changing signals. Once the feature extraction within the analysis side chain is completed, then the features from each channel are analyzed in order to determine new panning positions for each channel. Control signals are sent to the signal processing side in order to trigger the desired panning commands.

3.2. Adaptive Gating. Because noise on an input microphone channel may trigger undesired readings, the input signals are gated. The threshold of the gate is determined in an adaptive manner. By noise we refer not only to random ambient noise but also to interference due to nearby sources, such as the sound from adjacent instruments that are not meant to be input to a given channel.

Adaptive gating is used to ensure that features are extracted from a channel only when the intended signal is present and significantly stronger than the noise sources. The gating method is based on a method implemented in [16, 17] by Dugan. A reference microphone may be placed outside of the usable source microphone area to capture a signal representative of the undesired ambient and interference

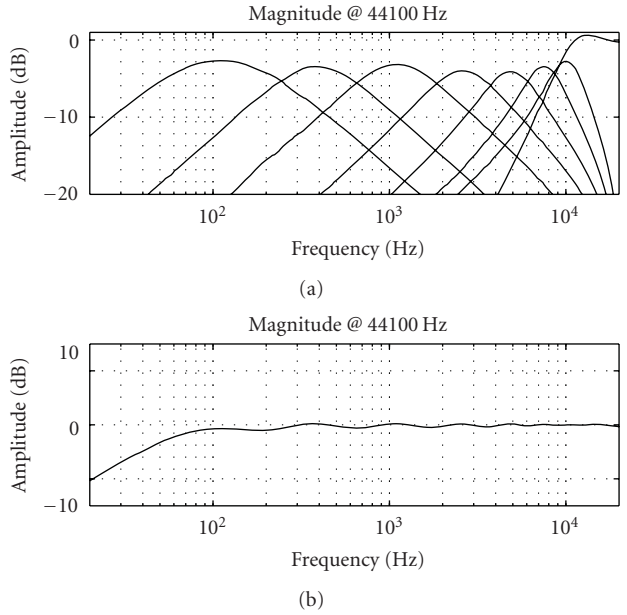


Figure 2: Quasiflat frequency response bandpass filter bank (Type A filter bank for $K = 8$). (a) Filter bank consisting of a set of eight second order band-pass IIR Biquadratic filters with center frequencies as follow: 100 Hz, 400 Hz, 1 kHz, 2.5 kHz, 5 kHz, 7.5 kHz, 10 kHz, and 15000 kHz. (b) Combined response of the filter bank.

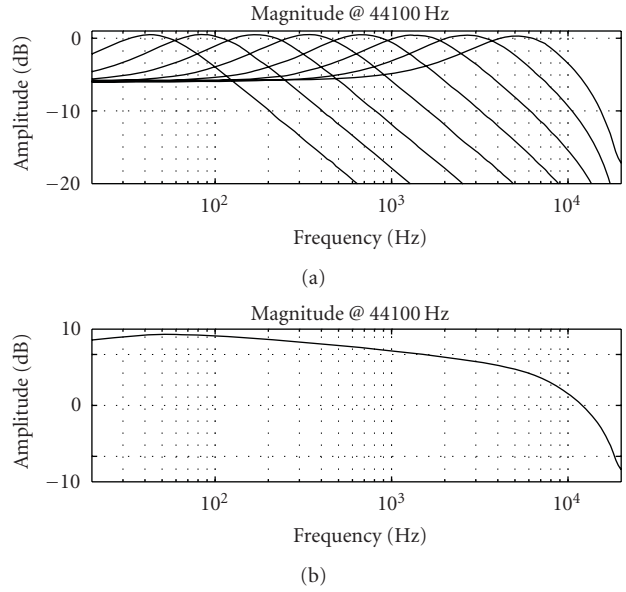


Figure 3: Lowpass filter decomposition filter bank (Type B filter bank for $\mathbf{K} = \mathbf{8}$). (a) Filter bank comprised of a set of second order low-pass IIR Biquadratic filters with cutoff frequencies as follows: 35 Hz, 80 Hz, 187.5 Hz, 375 Hz, 750 Hz, 1.5 kHz, 3 kHz, and 6 kHz. (b) Combined response of the filter bank. All gains have been set to have a maximum peak value of 0 dBs.

noise. The reference microphone signal is used to derive an adaptive threshold by opening the gate only if the input signal magnitude is grater than the reference microphone magnitude signal. Therefore the input signal is only passed to the side processing chain when its level exceeds that of the reference microphone signal.

3.3. Filter Bank Implementation. The implementation uses a filter bank to perform spectral decomposition of each individual channel. The filter bank does not affect the audio path since it is only used in the analysis section of the algorithm. It was chosen as opposed to other methods of classifying the dominant frequency or frequency range of a signal [18] because it does not require Fourier analysis, and hence is more amenable to a real-time implementation.

For the purpose of finding the optimal spectral decomposition for performing automatic panning, two different eight-band filter banks were designed and tested. The first consisted of a quasiflat frequency response bandpass filter bank, which for the purposes of this paper we will call filter bank type A in Figure 2, and the second contained a lowpass filter decomposition filter bank, which we will call filter bank type B in Figure 3. In order to provide an adaptive frequency resolution for each filter bank, the total number of filters, K, is equal to the number of input channels that are meant to be panned. The individual gains of each filter were optimized to achieve a quasiflat frequency response.

3.4. Determination of Dominant Frequency Range. Once the filter bank has been designed, the algorithm uses the band limited signal in each filter's output to obtain the absolute peak amplitude for each filter. The peak amplitude is measured within a 100 ms window. The algorithm uses the spectral output of each filter contained within the filter bank to calculate the peak amplitude of each band. By comparing these peak amplitudes, the filter with the highest peak is found. An accumulated score is maintained for the number of occurrences of the highest peak in each filter contained within the filter bank. This results in a classifier that determines the dominant filter band for an input channel from the highest accumulated score.

The block diagram of the filter bank analysis algorithm is provided in Figure 4. It should be noted that this approach uses digital logic operations of comparison and addition only, which makes it highly attractive for an efficient digital implementation.

3.5. Cross-Adaptive Mapping. Now that each input channel has been analyzed and associated with a filter, it remains to define a mapping which results in the panning position of each output channel. The rules which drive this cross-adaptive mapping are as follows.

The first rule implements the constraint that low-frequency sources should not be panned. Thus, all sources with accumulated energy contained in a filter with a high cutoff frequency below 200 Hz are not panned and remain centered at all times.

The second rule decides the available panning step of each source. This is a positioning rule which uses equidistant spacing of all sources with the same dominant frequency

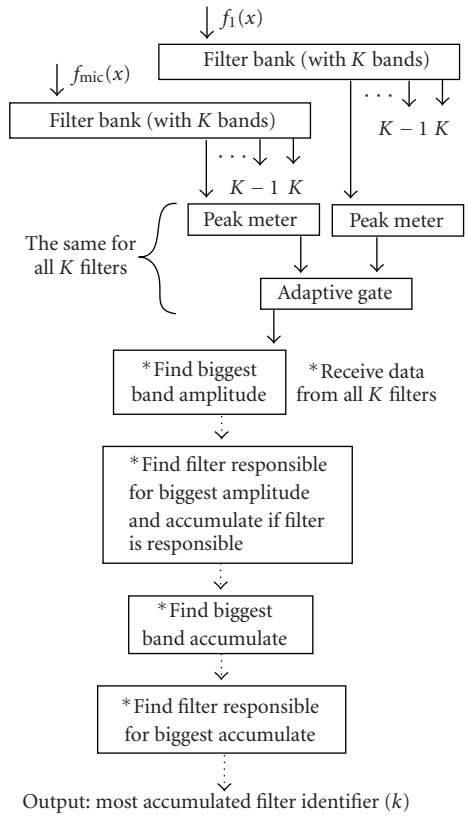


Figure 4: Analysis block diagram for one input channel.

range. Initially, all sources are placed in the centre. The available panning steps are calculated for every different accumulated filter, k, where k ranges from 1 to K, based on the number of sources, N_k, whose dominant frequency range resides in that filter. Due to this channel dependency the algorithm will update itself every time a new input is detected in a new channel, or if the spectral content of an input channel suffers from a drastic change over time.

For filters which have not reached maximum accumulation there is no need to calculate the panning step, which makes the algorithm less computationally expensive. If only one repetition exists for a given kth filter ($N_k = 1$) the system places the input at the center.

The following equation gives the panning space location of the ith source residing in the kth filter:

$$P(i,k) = \begin{cases} 1/2, & N_k = 1, \\ \dfrac{N_k - i - 1}{2(N_k - 1)}, & i + N_k \text{ is odd}, \\ 1 - \dfrac{N_k - i}{2(N_k - 1)}, & i + N_k \text{ is even}, N_k \neq 1, \end{cases} \tag{1}$$

where $P(i,k)$ is the ith available panning step in the kth filter, i ranges from 1 to N_k and $P(i,k) = 0.5$ corresponds to a center panning position.

Using this equation, if N_k is odd, the first source, $i = 1$, is positioned at the center. When N_k is even, the first two sources, $i = 2, 3$, are positioned either side of the center. In both cases, the next two sources is positioned either side of the previous sources and so on such that sources are positioned further away from the centre as i increases. The extreme panning positions are 0 and 1. However, as mentioned, hard panning is generally not preferred. So our current implementation provides a panning width control, P_W, used to narrow or widen the panning range. The panning with has a valid range from 0 to 0.5 where 0 equates to the wide panning possible and 0.5 equates to no panning. In our current implementation, it defaults to $P_W = 0.059$. The P_W value is subtracted for all panning positions bigger than 0.5 and added to all panning positions smaller than 0.5. In order to avoid sources originally panned left to cross to the right or sources originally panned right to cross to the left, the panning width algorithm ensures that sources in such cases default to the centre position.

3.6. Priority. Equation (1) provides the panning position for each of the sources with dominant spectral content residing in the kth filter, but it does not say how each of those sources are ordered from source $i = 1$ to $i = N_k$. The common practices mentioned earlier would suggest that certain sources, such as lead vocals, would be less likely to be panned to extremes than others, such as incidental percussions. However, the current implementation of our automatic panner does not have access to such information. Thus, the authors have proposed to use a priority driven system in which the user can label the channels according to importance. In this sense, it is a semiblind automatic system. Thus, all sources are ordered from highest to the lowest priority. For the N_k sources residing in the kth filter, the first panning step is taken by the highest priority source, the second panning step by the next highest priority source, and so on. The block diagram containing the constrained decision control rule stage of the algorithm is presented in Figure 5.

3.7. The Panning Processing. Once the appropriate panning position was determined, a sine-cosine panning law [2] was used to place sources in the final sound mix:

$$\begin{aligned} f_{\text{Rout}}(x) &= \sin\left(\frac{P(i,k)\pi}{2}\right) \cdot f_{\text{in}}(x), \\ f_{\text{Lout}}(x) &= \cos\left(\frac{P(i,k)\pi}{2}\right) \cdot f_{\text{in}}(x). \end{aligned} \tag{2}$$

An interpolation algorithm has been coded into the panner to avoid rapid changes of signal level. The interpolator has a 22 ms fade-in and fade-out, which ensures a smooth natural transition when the panning control step is changed.

In Figure 6, the result of down-mixing 12 sinusoidal test signals through the automatic panner is shown. It can be seen that both f_1 and f_{12} are kept centered and added together because their spectral content is below 200 Hz. The three sinusoids with a frequency of 5 kHz have been evenly spread.

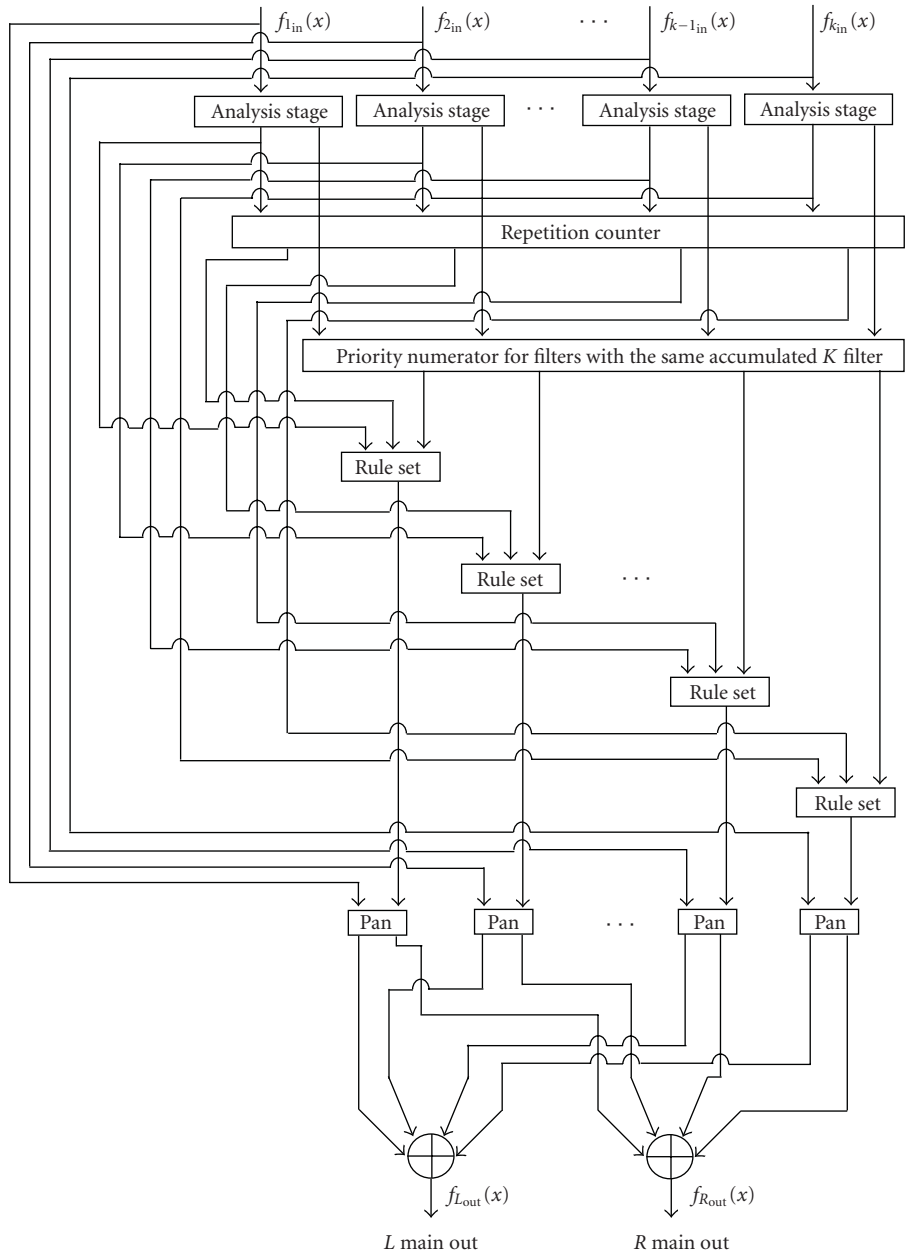


Figure 5: Block diagram of the automatic panner constrained control rules algorithm for K input channels.

f_2 has been allocated to the center due to priority while f_4 has been send to the left and f_6 has been send to the right, in accordance with (1). Because there is no other signal with the same spectral content than f_{11}, it has been assigned to the center. The four sinusoids with a spectral content of 15 kHz have been evenly spread. Because of priority, f_3 has been assigned a value of 0.33, f_7 has been assigned a value of 0.66, f_9 has been assigned all the way to the left, and f_{10} has been assigned all the way to the right, in accordance with (1). Finally, the two sinusoids with a spectral content of 20 KHz have been panned to opposite sides. All results prove to be in accordance with the constraint rules proposed for cross-adaptive mapping.

4. Results

4.1. Test Design. In order to evaluate the performance of the semiautonomous panner algorithm against human performance, a double blind test was designed. Both of auto-panning algorithms were tested, the bandpass filter classifier known as algorithm type A, and the low-pass classifier known as algorithm type B. Algorithms were randomly tested in a double blind fashion.

The control group consisted of three professional human experts and one nonexpert, who had never panned music before. The test material consisted of 12 multitrack songs of different styles of music. Stereo sources were used in the form

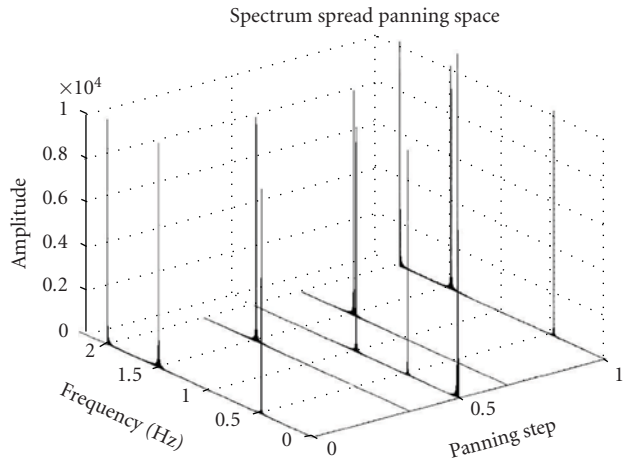


Figure 6: Results of automatic panning based on the proposed design. The test inputs were 12 sinusoids with amplitude equal to one and the following frequencies: $f_1 = 125\,\text{Hz}$, $f_2 = 5\,\text{kHz}$, $f_3 = 15\,\text{kHz}$, $f_4 = 5\,\text{kHz}$, $f_5 = 20\,\text{kHz}$, $f_6 = 5\,\text{kHz}$, $f_7 = 15\,\text{kHz}$, $f_8 = 20\,\text{kHz}$, $f_9 = 15\,\text{kHz}$, $f_{10} = 15\,\text{kHz}$, $f_{11} = 10\,\text{kHz}$, and $f_{12} = 125\,\text{Hz}$.

of two separate monotracks. Where acoustic drums were used, they would be recorded with multiple microphones and then premixed down into a stereo mix. Humans and algorithms used the same stereo drum and keyboard tracks as separate left and right monofiles. All 12 songs were panned by the expert human mixers and by the nonexpert human mixer. They were asked to pan the song while listening for the first time. They had the length of the song to determine their definitive panning positions. The same songs were passed through algorithms A and B only once for the entire length of the song. Although the goal was to give the human and machine mixers as close to the same information as possible, human mixers had the advantage of knowing which type of instrument it was. Therefore, they assigned priority according to this prior known knowledge. For this reason a similar priority scheme was chosen to compensate for this. Both A and B algorithms used the same priority schema. Mixes used during the test contain music freely available under creative commons copyright can be located in [19].

As shown in Figure 7, the test used two questions to measure the perceived overall quality of the panning for each audio comparison. For the first question, "how different is the panning of A compared to B?", a continuous slider with extremities marked "exactly the same" and "completely different" was used. The answer obtained in this question was used as a weighting factor in order to decide the validity of the next question. The second question, "which file, A or B, has better panning?", used a continuous slider with extremes marked "A quality is ideal" and "B quality is ideal". For both of these questions, no visible scale was added in order not to influence their decision. The test subjects were also provided with a comment box that was used for them to justify their answers to the previous two questions. During the test it was observed that expert subjects tend to use the name of the instrument to influence their panning decisions. In other words they would look for the "bass" label to make sure that they kept it center. This was an encouraging sign that panning amongst professionals follows constraint rules similar to those that were implemented in the algorithms.

The tested population consisted of 20 professional sound mixing engineers, with an average experience of 6-year work in the professional audio market. The tests were performed over headphones, and both the human mixers and the test subjects used exactly the same headphones. The test lasted an average time of 82 minutes.

Double blind A/B testing was used with all possible permutations of algorithm A, algorithm B, expert and amateur panning. Each tested user answered a total of 32 questions, two of which were control questions, in order to test the subject's ability to identify stereo panning. The first control question consisted of asking the test subjects to rate their preference between a stereo and a monaural signal. During the initial briefing it was stressed to the test subject that stereo is not necessarily better than monaural audio. The second control question compared two stereo signals that had been panned in exactly the same manner.

4.2. Result Analysis. All resulting permutations were classified into the following categories: monaural versus stereo, same stereo versus same stereo file, method A versus method B, method A versus nonexpert mix, method B versus nonexpert mix, method A versus expert mix, and method B versus expert mix.

Results obtained on the question "How different is panning A compared to B?" were used to weight the results obtained for the second question "Which file, A or B, has better panning quality?". This is in order to have a form of neglecting incoherent answers such as "I find no difference between files A or B but I find the quality of B to be better compared to A".

Answers to the first question showed that, with at least 95% confidence, the test subjects strongly preferred stereo to monaural mixes. The second question also confirmed with at least 95% confidence that professional audio engineers find no significant difference when asked to compare two identical stereo tracks. The results are summarized in Table 1, and the evaluation results with 95% confidence intervals are depicted in Figure 8.

The remaining tests compared the two panning techniques against each other and against expert and nonexpert mixes. The tested audio engineers preferred the expert mixes to panning method A, but this result could only be given with 80% confidence. On average, non-expert mixes also were preferred to panning method A, but this result could not be considered significant, even with 80% confidence.

In contrast, panning method B was preferred over nonexpert mixes with over 90% confidence. With at least 95% confidence, we can also state that method B was preferred over method A. Yet when method B is compared against expert mixes, there is no significant difference.

The preference for panning method B implies that low-pass spectral decomposition is preferred over band-pass spectral decomposition as a means of signal classification for the purpose of semi-autonomous panning. Furthermore, the

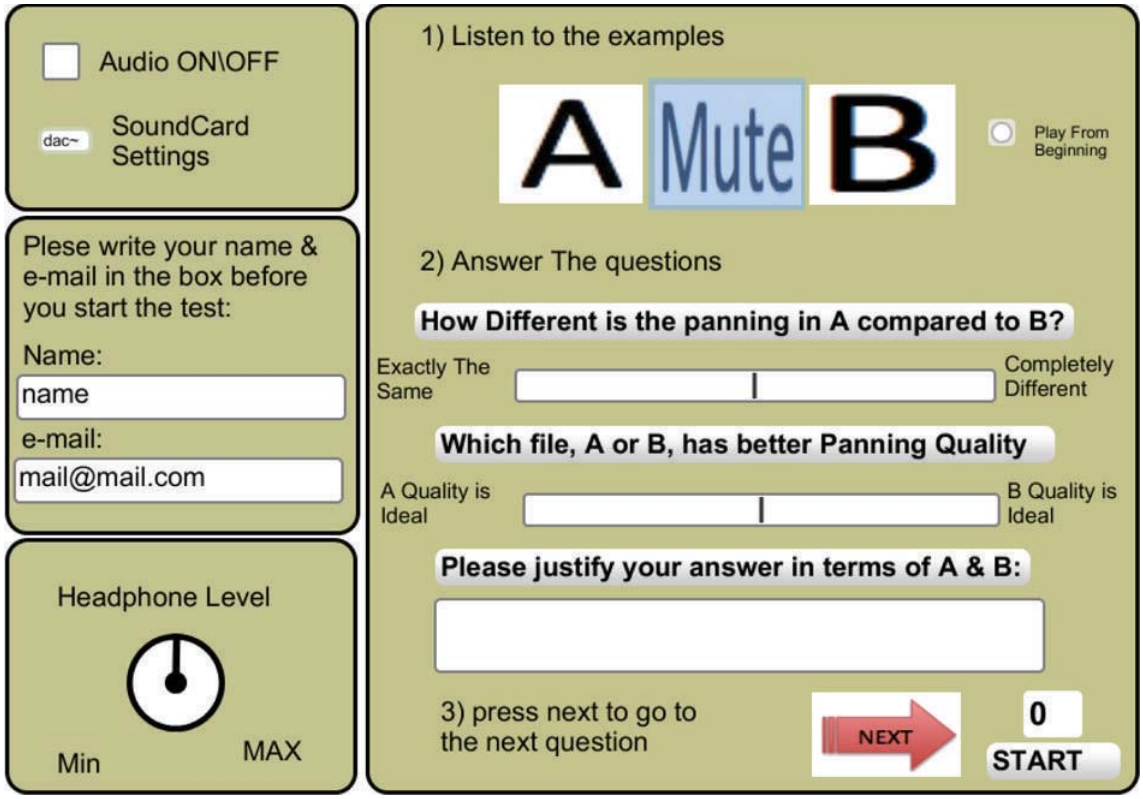


Figure 7: Double blind panning quality evaluation test interface.

lack of any statistical difference between panning method B and expert mixe, (in contrast to the significant preference for method B over non-expert mixes, and for expert mixes over method A) leads us to conclude that the semi-autonomous panning method B performs roughly equivalently to an expert mixing engineer.

It was found that the band-pass filter bank, method A, tended to assign input channels to less filters than the low-pass filter bank, method B. The distribution of input tracks among filters for an 8-channel song for both methods is depicted in Figure 9. In effect, panning method B is more discriminating as to whether two inputs have overlapping spectral content and hence is less likely to unnecessarily place sources far from each other. This may account for the preference of panning method B over panning method A.

The subjects justified their answers in accordance with the common practices mentioned previously. They relied heavily on manual instrument recognition to determine the appropriate position of each channel. It was also found that any violation of common practice, such as panning the lead vocals, would result in a significantly low measure of panning quality. One of the most interesting findings was that spatial balance seemed to be not only a significant cue used to determine panning quality but was also a distinguishing factor between expert and non-expert mixes. Non-expert mixes were often weighted to one side, whereas almost universally, expert mixes had the average source position in the centre. Both panning methods A and B were devised to perform optimal left to right balancing. Histograms of source positions which demonstrate these behaviors are depicted in Figure 10.

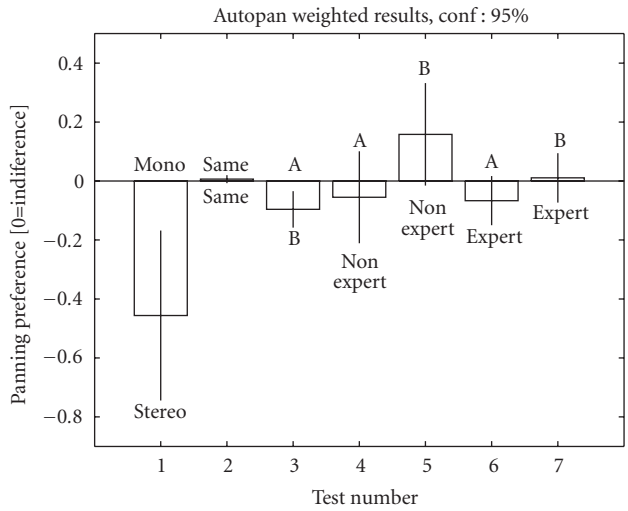


Figure 8: Summarized results for the subjective evaluation. The first two tests were references (comparing stereo against monaural, and comparing identical files), and the remaining questions compared the two proposed auto-panning methods against each other and against expert and non-expert mixes. 95% confidence intervals are provided.

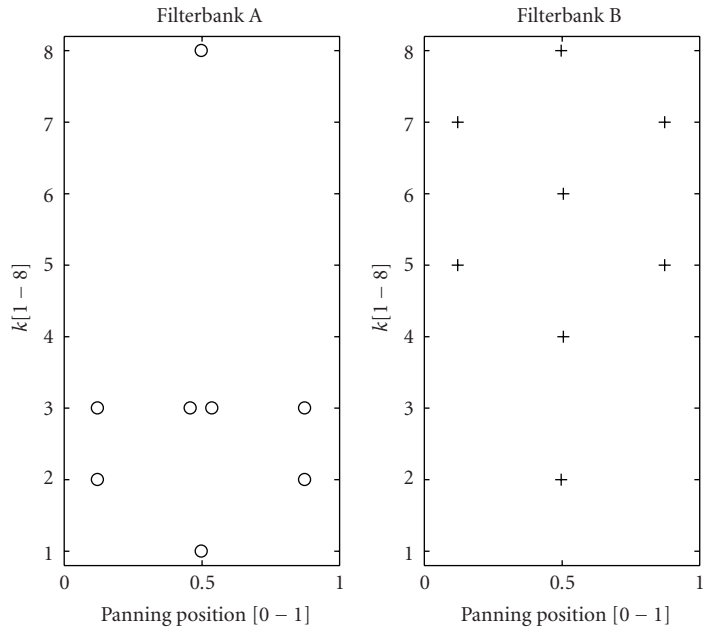


Figure 9: Distribution of sources among filters for methods A and B for the same song ($P_W = 0.059$).

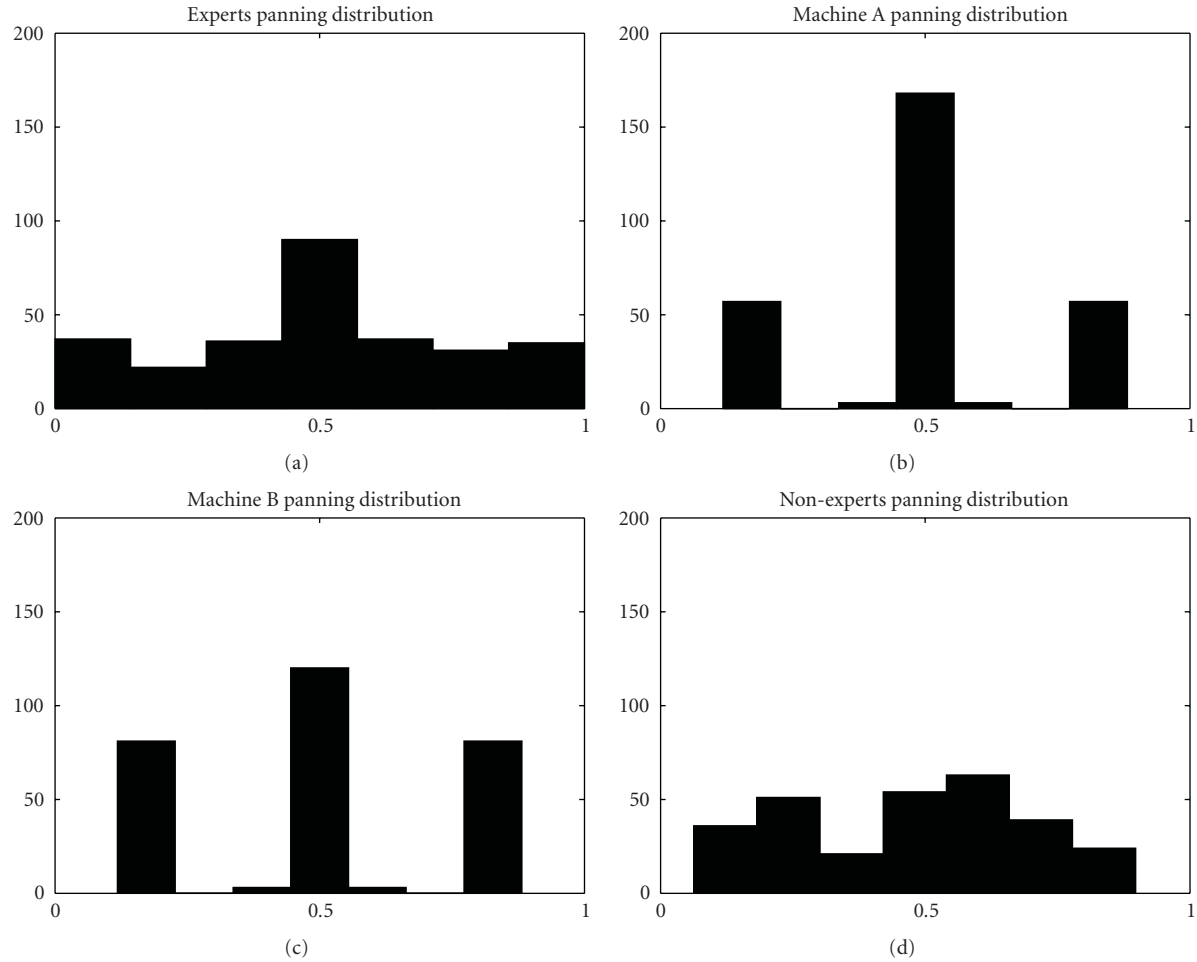


Figure 10: Panning space distribution histograms ($P_W = 0.059$).

Table 1: Double blind panning quality evaluation table.

Test	Number of Comparisons	Preference	Confidence	Standard Deviation	Mean
Stereo versus Mono	20	Stereo	95%	0.61545	−0.4561
Stereo versus Stereo	20	Identify them to be the same	95%	0.0287	0.0064
Human Expert versus Method A	144	Human	80%	0.5105	−0.666
Method A versus Non-expert	56	No significant difference between algorithms	95%	0.5956	−0.0552
Method B versus Non-expert	56	Method B	90%	0.6631	0.1583
Method B versus Expert	144	No significant difference between algorithms	95%	0.5131	0.0108
Method A versus Method B	200	Method B	95%	0.4474	−0.0962

5. Conclusions and Future Work

In terms of generating blind stereo panning up-mixes with minimum human interactions, we can conclude that it is possible to generate intelligent expert systems capable of performing better than a non-expert human while having no statistical difference when compared to a human expert. According to the subjective evaluation, low-pass filter-bank accumulative spectral decomposition features seem to perform significantly better than band-pass decompositions.

More sophisticated forms of performing source priority identification in an unaided manner need to be investigated. To further automate the panning technique, instrument identification and other feature extraction techniques could be employed to identify those channels with high priority. Better methods of eliminating microphone cross-talk noise need to be researched. Furthermore, in live sound situations, the sound engineer would have visual cues to aid in panning. For instance, the relative positions of the instruments on stage are often used to map sound sources in the stereo field. Video analysis techniques could be used to incorporate this into the panning constraints. Future subjective tests should include visual cues and be performed in real sound reinforcement conditions.

Finally, the work presented herein was restricted to stereo panning. In a two-or three-dimensional sound field, there are more degrees of freedom, but rules still apply. For instance, low-frequency sources are often placed towards the ground, while high-frequency sources are often placed above, corresponding to the fact that high frequency sources emitted near or below the ground would be heavily attenuated [20]. It is the intent of the authors to extend this work to automatic placement of sound sources in a multispeaker, spatial audio environment.

References

[1] M. A. Gerzon, "Signal processing for simulating realistic stereo images," in *Proceedings of the 93rd Convention Audio Engineering Society*, San Francisco, Calif, USA, October 1992.

[2] J. L. Anderson, "Classic stereo imaging transforms—a review," submited to *Computer Music Journal*, http://www.dxarts.washington.edu/courses/567/08WIN/JL_Anderson_Stereo.pdf.

[3] D. Griesinger, "Stereo and surround panning in practice," in *Proceedings of the 112th Audio Engineering Society Convention*, Munich, Germany, May 2002.

[4] E. Perez_Gonzalez and J. Reiss, "Automatic mixing: live down-mixing stereo panner," in *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx '07)*, pp. 63–68, Bordeaux, France, 2007.

[5] D. Self, et al., "Recording consoles," in *Audio Engineering: Know It All*, D. Self, Ed., vol. 1, chapter 27, pp. 761–807, Newnes/Elsevier, Oxford, UK, 1st edition, 2009.

[6] R. Neiman, "Panning for gold: tutorials," *Electronic Musician Magazine*, 2002, http://emusician.com/tutorials/emusic_panning_gold/.

[7] R. Izhaki, "Mixing domains and objectives," in *Mixing Audio: Concepts, Practices and Tools*, chapter 6, pp. 58–71, Focal Press/Elsevier, Burlington, Vt, USA, 1st edition, 2007.

[8] R. Izhaki, "Panning," in *Mixing Audio: Concepts, Practices and Tools*, chapter 13, pp. 184–203, Focal Press/Elsevier, Burlington, Vt, USA, 1st edition, 2007.

[9] B. Bartlett and J. Bartlett, "Recorder-mixers and mixing consoles," in *Practical Recording Techniques*, chapter 12, pp. 259–275, Focal Press/Elsevier, Oxford, UK, 3rd edition, 2009.

[10] B. Owsinski, "Element two: panorama—placing the sound in the soundfield," in *The Mixing Engineer's Handbook*, chapter 4, pp. 20–24, Mix Books, Vallejo, Calif, USA, 2nd edition, 2006.

[11] F. Rumsey and T. McCormick, "Mixers," in *Sound and Recording: An Introduction*, chapter 5, pp. 96–153, Focal Press / Elsevier, Oxford, UK, 1st edition, 2006.

[12] P. White, "The creative process: pan position," in *The Sound on Sound Book of Desktop Digital Sound*, pp. 169–170, MPG Books, UK, 1st edition, 2000.

[13] E. Benjamin, "An experimental verification of localization in two-channel stereo," in *Proceedings of the 121st Convention Audio Engineering Society*, San Fransisco, Calif, USA, 2006.

[14] J. Beament, "The direction-finding system," in *How we Hear Music: The relationship Between Music and the Hearing Mechanism*, pp. 127–130, The Boydel Press, Suffolk, UK, 2001.

[15] V. Verfaille, U. Zölzer, and D. Arfib, "Adaptive Digital Audio Effects (A-DAFx): a new class of sound transformations," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1817–1831, 2006.

[16] D. W. Dugan, "Automatic microphone mixing," *Journal of the Audio Engineering Society*, vol. 23, no. 6, pp. 442–449, 1975.

[17] D. W. Dugan, "Application of automatic mixing techniques to audio consoles," in *Proceedings of the 87th Convention of the Audio Engineering Society*, pp. 18–21, New York, NY, USA, October 1989.

[18] W. A. Sethares, A. J. Milne, S. Tiedje, A. Prechtl, and J. Plamondon, "Spectral tools for dynamic tonality and audio morphing," *Computer Music Journal*, vol. 33, no. 2, pp. 71–84, 2009.

[19] E. Perez_Gonzalez and J. Reiss, "Automatic mixing tools for audio and music production," 2010, http://www.elec.qmul.ac.uk/digitalmusic/automaticmixing/.

[20] D. Gibson and G. Peterson, *The Art of Mixing: A Visual Guide to Recording, Engineering and Production*, Mix Books / ArtistPro Press, USA, 1st edition, 1997.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 642316, 18 pages
doi:10.1155/2010/642316

Research Article

Two-Dimensional Beam Tracing from Visibility Diagrams for Real-Time Acoustic Rendering

F. Antonacci (EURASIP Member), A. Sarti (EURASIP Member), and S. Tubaro (EURASIP Member)

Dipartimento di Elettronica ed Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

Correspondence should be addressed to F. Antonacci, antonacc@elet.polimi.it

Received 26 February 2010; Revised 24 June 2010; Accepted 25 August 2010

Academic Editor: Udo Zoelzer

We present an extension of the fast beam-tracing method presented in the work of Antonacci et al. (2008) for the simulation of acoustic propagation in reverberant environments that accounts for diffraction and diffusion. More specifically, we show how visibility maps are suitable for modeling propagation phenomena more complex than specular reflections. We also show how the beam-tree lookup for path tracing can be entirely performed on visibility maps as well. We then contextualize such method to the two different cases of channel (point-to-point) rendering using a headset, and the rendering of a wave field based on arrays of speakers. Finally, we provide some experimental results and comparisons with real data to show the effectiveness and the accuracy of the approach in simulating the soundfield in an environment.

1. Introduction

Rendering acoustic sources in virtual environments is a challenging problem, especially when real-time operation is required without giving up a realistic impression of the result. The literature is rich with methods that approach this problem for a variety of purposes. Such methods are roughly divided into two classes: the former is based on an approximate solution of the wave equation on a finite grid, while the latter is based on the geometric modeling of acoustic propagation. Typical examples of the first class of methods are based on the solution of the Green's or Helmholtz-Kirchoff's equation through finite and boundary element methods [1–3]. The computational effort required by the solution of the wave equation, however, makes these algorithms unsuitable for real-time operation except for a very limited range of frequencies. Geometric methods, on the other hand, are the most widespread techniques for the modeling of early acoustic reflections in complex environments. Starting from the spatial distribution of the reflectors, their acoustic properties, and the location and the radiation characteristics of sources and receivers (listening points), geometric methods cast rays in space and track their propagation and interaction with obstacles in the environment [4]. The sequence of reflections, diffractions and diffusions a ray undergoes constitutes the acoustic path that link source and receiver.

Among the many available geometric methods, a particularly efficient one is represented by beam tracing [5–9]. This method was originally conceived by Hanrahan and Heckbert [5] for applications of image rendering, and was later extended by Funkhouser et al. [10] to the problem of audio rendering. A beam is intended as a bundle of acoustic rays originating from a point in space (a real source or a wall-reflected one), which fall onto the same planar portion of an acoustic reflector. Every time a beam encounters a reflector, in fact, it splits into a set of subbeams, each corresponding to a different planar region of that reflector or of some other reflector. As they bounce around in the environment, beams keep branching out. The beam-tracing method organizes and encodes this beam splitting/branching process into a specialized data structure called *beam-tree*, which describes the information of the visibility of a region from a point (i.e., the source location). Once the beam-tree is available, path-tracing becomes a very efficient process. In fact, given the location of the listening point (receiver), we can immediately determine which beams illuminate it, just through a "look up" of the beam-tree data structure. We should notice,

however, that with this solution the computational effort associated to the beam tracing process and that of path-tracing are quite unbalanced. In fact if the environment is composed by n reflectors, the exhaustive test of the mutual visibility among all the n reflectors involves $\mathcal{O}(n^3)$ tests, while the test of the presence of the receiver in the m traced beams needs only $\mathcal{O}(m)$ tests. Some solutions for a speedup of the computation of the beam-tree have been proposed in the literature. As an example in [10] the authors adopt the Binary Space Partitioning Technique to operate a selection of the visible obstacles from a prescribed reflector. A similar solution was recently proposed in [11], where the authors show that a real-time tracing of acoustic paths is possible even in a simple dynamic environment.

In [12] the authors generalized traditional beam tracing by developing a method for constructing the beam-tree through a lookup on a precomputed data structure called global visibility function, which describes the visibility of a region not just as a function of the viewing angle but also of the source location itself.

Early reflections are known to carry some information on the geometry of the surrounding space and on the spatial positioning of acoustic sources. It is in the initial phase of reverberation, in fact, that we receive the echoes associated to the first wall reflections. Other propagation phenomena, such as diffusion, transmission and diffraction tend to enrich the sense of presence in "virtual walkthrough" scenarios, especially in densely occluded environments. As beam tracing was originally conceived for the modeling of specular reflections only, some extensions of this method were proposed to account for other propagation phenomena. Funkhouser et al. [13], for example, account for diffusion and diffraction through a bidirectional beam tracing process. When the two beam-trees that originate from the receiver and the source intersect on specific geometric primitives such as edges and reflectors, propagation phenomena such as diffusion and diffraction could take place. The need of computing two beam-trees, however, poses problems of efficiency when using conventional beam tracing methods, particularly when sources and/or receivers are in motion. A different approach was proposed by Tsingos et al. [14], who proposed to use the uniform theory of diffraction (UTD) [15] by building secondary beam-trees originated from the diffractive edges. This approach is quite efficient, as the tracing of the diffractive beam-trees can be based on the sole geometric configuration of reflectors. Once source and receiver locations are given, in fact, a simple test on the diffractive beam-trees determines the diffractive paths. Again, however, this approach inherits the advantages of beam tracing but also its limits, which are in the fact that a new beam-tree needs be computed every time a source moves.

As already mentioned above, in [12] we proposed a method for generating a beam-tree through a lookup on the global visibility function. That method had the remarkable advantage of computing a large number of acoustic paths in real time as both source and reflector are in motion in a complex environment. In this paper we generalize the work proposed in [12] in order to accommodate diffusion and diffraction phenomena. We do so by revisiting the concept of global visibility and by introducing novel lookup methods and new operators. Thanks to these generalizations, we will also show how it is possible to work on the visibility diagrams not just for constructing beam-trees but also to perform the whole path-tracing process.

In this paper we expand and repurpose the beam tracing method for applications of real-time rendering of acoustic sources in virtual environments. Two are the envisioned scenarios: in the former the user is wearing a headset, in the latter the whole sound field within a prescribed volume is rendered using loudspeaker arrays. We will show that the two scenarios share the same beam tracing engine which, in the first case, is followed by a path-tracing algorithm based on beam-tree lookup [12], with an additional head-related transfer function. In the second case the beam tracer is used for generating the control parameters of the beam-shaping algorithm proposed in [16]. This beam-shaping method allows us to design the spatial filter to be applied to the loudspeaker arrays for the rendering of an arbitrary beam. Other solutions exist in the literature for the rendering of virtual environments, such as wave field synthesis (WFS) and ambisonics. Roughly speaking, WFS computes the spatial filter to be applied to the speakers with an approximation of the Helmholtz-Kirchoff's equation. Interestingly enough, for example, in [17] the task of computing the parameters of all the virtual sources in the environment is demanded to an image-source algorithm. Therefore, some WFS systems already partially rely on geometric methods. When rendering occluded environments, however, the image-source method tends to become computationally demanding, while fast beam tracing techniques [12] can offer a significant speedup.

It is important to notice that the method proposed in [12] was developed for modeling complex acoustic reflections in a specific class of 3D environments obtained as the cartesian product between a 2D floor plan and a 1D (vertical) direction. This situation, for example, describes a complex distribution of vertical walls ending in horizontal floor and ceiling. When considering acoustic wall transmission, a $2D \times 1D$ environment becomes useful for modeling a multi-floored building with a repeated floor plan. Although $2D \times 1D$ environments enjoy the advantages of 2D modeling (simplicity, duality, etc.), the computation of all delays and path lengths still needs to be performed in a 3D space. While this is rather straightforward in the case of geometric reflections, it becomes more challenging when dealing with diffraction and diffusion phenomena.

The paper is organized as follows. In Section 2 we review and revisit the concept of global visibility and its use for efficiently tracing acoustic paths. In Section 3 we discuss the main mathematical models used for explaining diffusion and diffraction phenomena, and we choose the one that best suits our beam tracing approach. Sections 4 and 5 focus on the modeling of diffusion and diffraction with visibility diagrams. In Section 6 we present two possible applications of the algorithm presented in this paper. In Section 7 we prove the efficiency and the effectiveness of our modeling solution. Finally, Section 8 provides some final comments and conclusions.

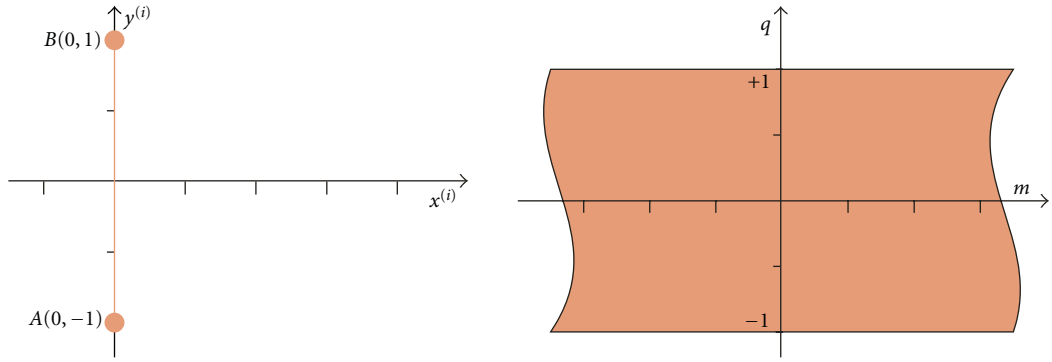


Figure 1: The specialized RRP (a) and the set of rays passing through the reference reflector in the (m, q) domain (visibility region from the reference reflector) (b).

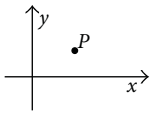
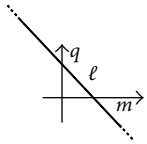
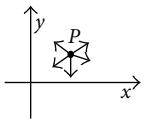
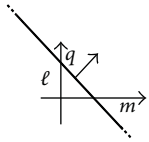
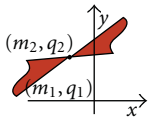
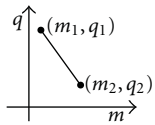
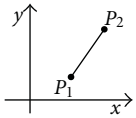
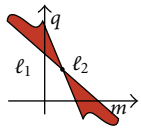
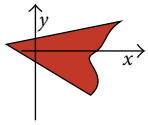
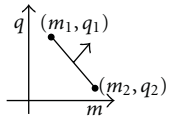
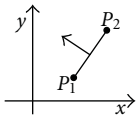
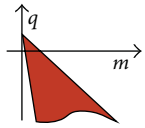
2. The Visibility Diagram Revisited

In this section we review the concept of visibility diagram, as it is a key element for the remainder of this paper. In [12] we adopted this representation for generating a specialized data structure that could swiftly provide information on how to trace acoustic beams and rays in real time with the rules of specular reflection. This approach constitutes a generalization of the beam tracing algorithm proposed by Hanrahan and Heckbert [5]. The visibility diagram is a remapping of the geometric structures and functional elements that constitute the geometric world (rays, beams, reflectors, sources, receivers, etc.) onto a special parameter space that is completely dual to the geometric one. Visibility diagrams are particularly useful for immediately assessing what is in the line of sight from a generic location and direction in space. We will first recount the basic concepts of visibility diagrams and provide a general view of the path-tracing problem for the specific case of purely specular reflections. This overview will be provided in a slightly more general fashion than in [12], as all the algorithmic steps will be given with reference to visibility diagrams, and will constitute the starting point for the discussions in the following sections.

2.1. Visibility and the Tracing Problem. A ray in a 2D space is uniquely characterized by three parameters: two for the location of its origin, and one for its direction. As we are tracing paths during their propagation, we are interested in rays emerging from a reflector after bouncing off it. As a consequence, the origin corresponds to the virtual source. Furthermore, because we are interested in assessing only where the ray will end up, we can afford ignoring some information on where the ray is coming from, for example the source distance. This means that a ray description based on three parameters turns out to be redundant, and can be easily reduced to two parameters. In [12] we adopted the *Reference Reflector Parametrization* (RRP) parametrization based on the location of the intersection on the reference reflector and the travel direction of the ray. Although the RRP is referred to a frame attached to a specific reflector, this choice does not represent a limitation, due to the iterative nature of the visibility evaluation process. Let s_i be the reference reflector. For reasons that will be clearer later on, the RRP normalizes s_i through a translation, a rotation and a scaling of the axes in such a way that the reference reflector lies on the segment of the y-axis between -1 and 1. The set of rays passing through s_i is described by the equation $y^{(i)} = mx^{(i)} + q$. Figure 1 shows the reflector s_i referred to the normalized frame in the geometric domain (left). The set of rays passing through s_i is called region of visibility from s_i and it is represented by the horizontal strip (reference visibility strip) in the (m, q) domain. Due to the duality between primitives in (x, y) and (m, q) domains we will sometimes refer to the RRP as the *dual space*. We are interested in representing the mutual occlusions between reflectors in the dual space. With this purpose in mind, we split the visibility strip into *visibility regions*, each corresponding to the set of rays that hit the same reflector. According to the image-source principle, all the obstacles that lie in the same half space of the image-source, are discarded during the visibility test. As a convention, in the future we will use the rotation of the reference reflector which brings the image-source in the half-space $x^{(i)} < 0$. The above parameter space turns out to play a similar role as the dual of a geometric space. In Table 1 we summarize the representation of some geometric primitives in the parameter space. A complete derivation of the relations of Table 1 can be found in [12, 18]. Notice that the relation between primitives in the two domains is of complete duality. For example, the dual of the oriented reflector is a wedge in the (m, q) domain (sort of an oriented "beam" in parameter space). Conversely, the dual of an oriented beam (a single wedge in the (x, y) geometric space) is an oriented segment in the (m, q) domain (sort of an oriented "reflector" in parameter space).

2.1.1. Visibility Region. The parameters describing all rays originating from the reference reflector s_i form the region of visibility *from* that reflector. After normalization, this region takes on the strip-like shape described in Figure 1, which we refer to as "reference visibility strip". Those rays that originate

Table 1: Primitives in the geometric domain and their corresponding representation in ray space.

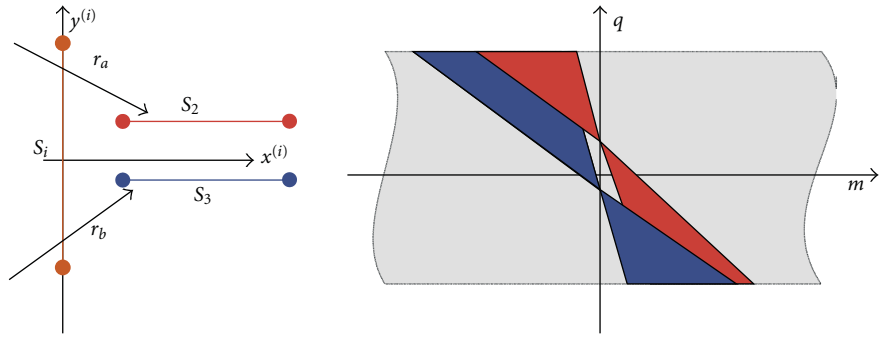
Geometric space	Ray space
Omnidirectional bundle of nonoriented rays 	Non-oriented ray or two-sided infinite reflector 
Omnidirectional bundle of outgoing rays (source) 	One-sided infinite reflector 
Beam (double wedge) 	Two-sided reflector 
Two-sided reflector 	Beam (double wedge) 
Oriented beam (single wedge) 	One-sided reflector 
One-sided reflector 	Oriented beam (single wedge) 

from the reference reflector and hit another reflector s_j form a subset of this strip (see Figure 1) which corresponds to the intersection between the dual of s_j and the dual of s_i (reference visibility strip). The intersection of the dual of s_j and the visibility strip is the visibility region of s_j from s_i. Once the source location is specified, the set of rays passing through s_j and s_i and departing from that location will be a subset of the visibility region of s_j. One key advantage of the visibility approach to the beam tracing problem resides in the fact that we only need geometric information about the environment to compute the visibility regions, which can therefore be computed in advance.

2.1.2. Dual of Multiple Reflectors: Visibility Diagrams. When there are more than two reflectors in the environment, we need to consider the possibility of mutual occlusions, which results in overlapping visibility regions. Sorting out which reflector occludes which (with respect to the reference reflector) corresponds to determining which visibility region *overrides* which in their overlap. Two solutions for the occlusion problem are possible: the first, already presented in [12], is based on a simple test in the geometric domain. An arbitrary ray chosen in the overlap of visibility regions can be cast to evaluate the front-to-back ordering of visibility regions or, more simply, to determine which oriented reflector is first met by the test ray. An example is provided in Figure 2 where, if s_i is the reference reflector, we end up having an occlusion between s_2 and s_3, which needs to be sorted out. A test ray is picked at random within the overlapping region to determine which reflector is hit first by the ray. This particular example shows that, unless we consider each reflector as the combination of two of oppositely-facing oriented reflectors, we cannot be sure that the occlusion problem can be disambiguated. In this case, for example, s_2 occludes s_3 for some rays, and s_3 occludes s_2 for others. As shown in Table 1, a two-sided reflector corresponds to a double wedge in ray space, each wedge corresponding to one of the two faces of the reflector. By considering the two sides of each reflector as individual oriented reflectors, we end up with four distinct wedge-like regions in ray space, thus removing all ambiguities. The overlap between visibility regions of two one-sided reflectors arises every time the extreme lines of the corresponding visibility regions intersect. We recall that the dual of a point $\overline{P}(\overline{x}, \overline{y})$ is a line whose slope is $-\overline{x}$. The extreme lines of the visibility region of reflector s_j are the dual of the endpoints of s_j, that are $(x_{j1}^{(i)}, y_{j1}^{(i)})$ and $(x_{j2}^{(i)}, y_{j2}^{(i)})$ and the slopes of the extreme lines of the visibility region of s_j are $-x_{ji}^{(i)}$ and $-x_{j2}^{(i)}$. A similar notation is used for the overlapping reflector s_k. Under the assumption that s_j and s_k never intersect in the geometric domain, we can reorder one-sided reflectors in front-to-back order by simply looking at the slopes of the extreme lines of their visibility regions. If the line $\ell_1^{(i)}$ of equation $q = -mx_{j1}^{(i)} + y_{j1}^{(i)}$ and the line $\ell_2^{(i)}$ of equation $q = -mx_{k2}^{(i)} + y_{k2}^{(i)}$ intersect in the dual space, then $-x_{j1}^{(i)} > -x_{k2}^{(i)}$ guarantees that s_j occludes s_k and $-x_{j1}^{(i)} < -x_{k2}^{(i)}$ guarantees that s_k occludes s_j.

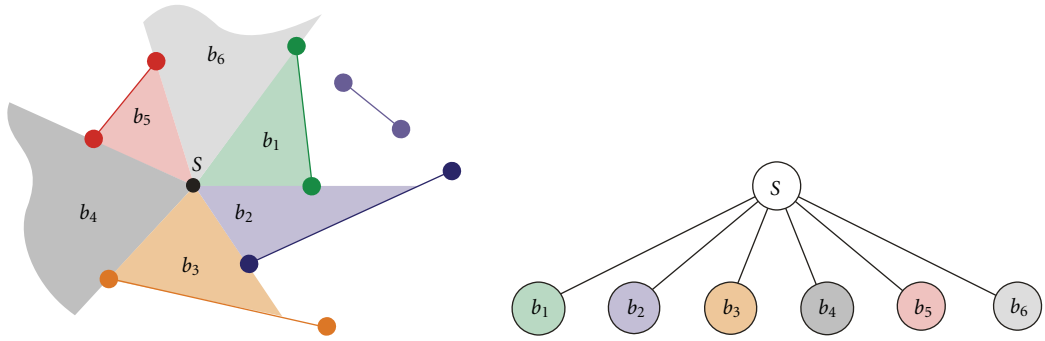
2.2. Tracing Reflective Beams and Paths in Dual Space

2.2.1. Tracing Beams. In this paragraph we summarize the tracing of beams in the geometric space using the information contained in the visibility diagrams. Further details on this specific topic can be found in [12]. This can be readily done by scanning the visibility diagram along the line that represents the "dual" of the virtual source. In fact, that line will be partitioned into a number of segments, one per visibility region. Each segment will correspond to a subbeam in the geometric space. Consider the configuration of reflectors of Figure 3(a). The first step of the algorithm consists of determining how the complete pencil of rays produced by the source S is partitioned into beams. This is done by evaluating the visibility from the source using traditional beam tracing. This initial splitting



(a) (b)

FIGURE 2: Ambiguity in the occlusion between two nonoriented reflectors s_2 and s_3. For some rays (e.g., r_a) s_2 occludes s_3. For other rays (e.g., r_b) s_3 occludes s_2.



(a) (b)

FIGURE 3: Beams traced from the source location (a) and the corresponding beam-tree (b).

process produces two classes of beams: those that fall on a reflector and those that do not. The beams and the corresponding beam-tree are shown in Figures 3(a) and 3(b), respectively. We consider the splitting of beam b_3, shown in Figure 4. The image-source is represented in the dual space by the line ℓ_P. The beam b_3 will therefore be a segment on that line, which will be partitioned in a number of segments, one for each region on the visibility diagram. In Figure 4(a) the beam splitting is accomplished in the (m, q) domain, while in Figure 4(b) we can see the corresponding subbeams in the geometric domain. This process is iterated for all the beams that fall onto a reflector. Further details can be found in [12]. At the end of the beam tracing process we end up with a tree-like data structure, each node b_k of which contains information that identifies the corresponding beam:

(i) the one-sided reference reflector s_i,

(ii) the one-sided illuminated reflector s_j (if any),

(iii) the position of the virtual source $S(x_s^{(i)}, y_s^{(i)})$ in the normalized reference frame,

(iv) the segment $[q_1, q_2]$ that identifies the "illuminating" region on the $y^{(i)}$-axis,

(v) the parent node (if any),

(vi) a list of the children nodes (if at least one exists).

The last two items are useful when reclaiming the "reflection history" of a beam. Given the above information we are immediately able to represent the beams (i.e., segments) in the (m, q) domain.

2.2.2. Tracing Paths. In [12] the construction of the beam-tree was accomplished in the dual space but path-tracing was entirely done in the geometric domain. We will now derive an alternate and more efficient procedure for tracing the acoustic paths directly in the dual space. The goal is to test the presence of the receiver R in the beam b_k, originating from the reflector s_i. The coordinates of the receiver in the normalized reference frame of s_i are $(x_r^{(i)}, y_r^{(i)})$. In order for R to be in b_k, there must exist a ray in b_k that passes through R, that is,

$$\exists (\overline{m}, \overline{q}) \in b_k : y_r^{(i)} = \overline{m} x_r^{(i)} + \overline{q}. \quad (1)$$

This means that the ray $(\overline{m}, \overline{q})$ from S to R, is represented in the dual space by a point resulting from the intersection of the dual of b_k (a segment) and the dual of R (a line). The presence test is thus performed by computing the intersection of two lines in the parameter space. If b_k does not fall onto a reflector, then the condition (1) is sufficient. If b_k falls on reflector s_j, then we must also make sure that s_j does not occlude R. Assuming that s_j lies on

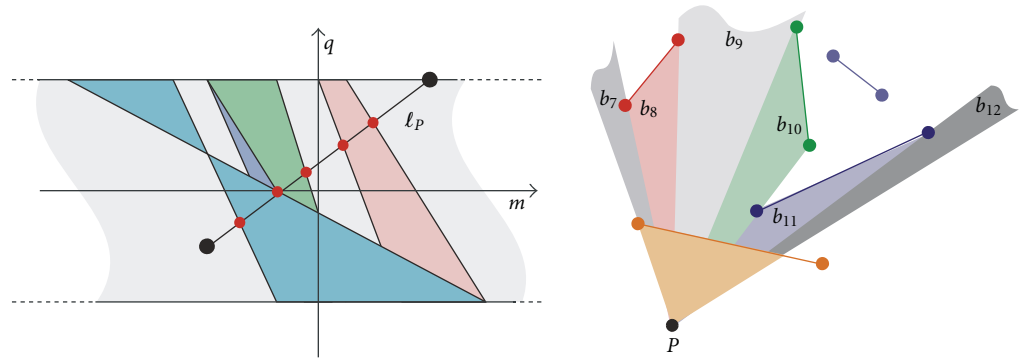


FIGURE 4: (a) Beam subdivision performed in the (m, q) domain for the bundle of rays corresponding to the reflection of the beam b_3 of Figure 3. (b) Corresponding subbeams in the geometric domain.

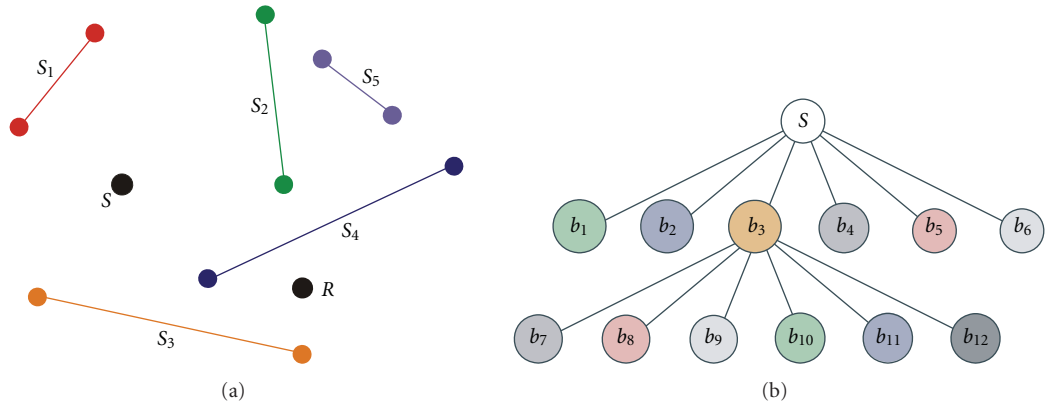


FIGURE 5: Path-tracing from source S to the receiver R. The beam-tree on the right-hand side is the same as in Figure 3.

the line $\ell_j^{(i)} : y = m_j x^{(i)} + q_j^{(i)}$, we can easily conclude that R is not occluded by s_j if the distance between S and R is smaller than the distance between S and the intersection of the $(\overline{m}, \overline{q})$ ray with s_j, which means that:

$$\begin{aligned}(x_s - x_r)^2 + (y_s - y_r)^2 \\ \leq \left(x_s - \frac{q_j - \overline{q}}{m_j - \overline{m}}\right)^2 \\ + \left(y_s - m_j \frac{q_j - \overline{q}}{m_j - \overline{m}} - q_j\right)^2.\end{aligned} \tag{2}$$

The conditions in (1) and (possibly) (2) are tested for all the beams. However, if R falls onto b_k, then we know that it cannot fall in other beams that share the virtual source of b_k. This speeds up the path-tracing process a great deal. As an example of the tracing process, consider the situation shown in Figure 5. Here S and R are not in the line of sight, but a reflective path exists through the reflection from s_3. First-order beams are traced directly in the geometric domain, as done in [12], therefore the presence of R in beams originating directly from S is tested directly in the geometric domain. Let us now test the presence of R in the reflected beams emerging from b_3 (see Figure 3). The intersection between

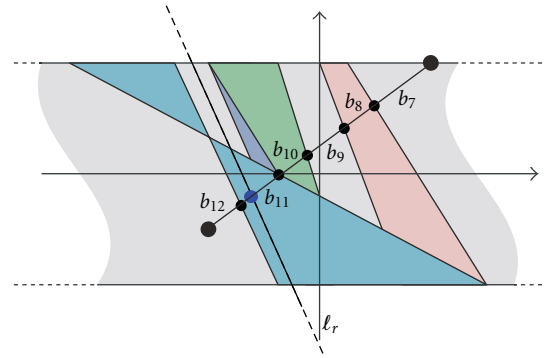


FIGURE 6: Tracing paths in the visibility diagram: the parameters of the outgoing ray are found by means of the intersection of the dual of the beam (a segment) and of the receiver (a line). Paths falling in beams limited by a reflector, also (2) must hold.

the line ℓ_r, dual of R, and the dual of b_{11} (a segment) is easily found (see Figure 6 on the right). Once we have checked the presence of the receiver in b_{11}, the position of the receiver and the information encoded in b_{11} are sufficient to determine the delay and the amplitude of the echo associated to that acoustic path. More details on this aspect will be provided in Section 6.1.

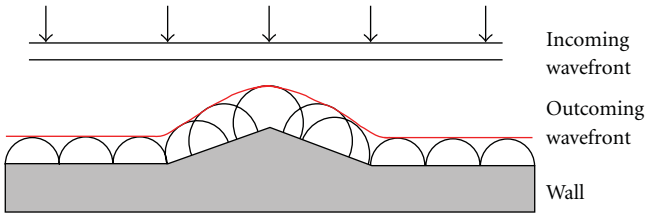


Figure 7: Diffusion phenomenon: a planar wavefront falls onto a rough surface from a perpendicular direction. The wavefront that bounces off the surface, resulting from a combination of secondary wavefronts, will not propagate just in the specular direction with respect to the exciting wave.

3. Mathematical Models of Diffraction and Diffusion

In this section we investigate some mathematical models used in the literature to quantitatively describe the causes of diffraction and diffusion. Later we will choose the model which best works for our beam tracing method.

3.1. Models of Diffusion. When a wavefront encounters a rough or nonhomogeneous surface, its energy is diffused in nonspecular directions (see Figure 7). Let us consider a flat surface with a single localized unevenness whose size is bigger than the wavelength λ of the incident wavefront. The Huygens principle interprets the diffused wavefront as the superposition of the local wavefronts associated to reflections on each point of the surface. As we can see in Figure 7, the direction of propagation of the outgoing wavefront differs from the direction of the incident one. Consequently, a sensor facing the wall will pick up energy not just from the incident wavefront but also from a direction that is not specular. A rough surface can be characterized through a statistical description of the speckles (in terms of size and density). In fact, the acoustic properties of the scattering material can be predicted or measured using various techniques [19–21]. Diffusion can also be associated to local variations of impedance (e.g., a flat reflective surface that exhibits areas of acoustically absorbing material) [22]. From the listener's standpoint, diffusion tends to greatly increase the number of paths between source and receiver and, consequently, the sense of presence [23]. Different models have been proposed in the literature to account for diffusion. A reflection is said to be totally diffusive (Lambertian) if the probability density function of the direction of the outgoing rays does not depend on the direction of the incoming ray. Totally diffused reflections are described by Lambert's cosine law. A survey on the typical acoustic characteristics of materials, however, reveals that Lambertian reflections turn out to be quite unrealistic. For this reason, in the literature we find two modeling descriptions: the *scattering coefficient* and the *diffusion coefficient* [24, 25]. The diffusion coefficient measures the similarity between the polar response of a Lambertian reflection and the actual one. This coefficient is expressed as the correlation index between the actual and the diffusive polar responses corresponding to a wavefront coming from a perpendicular direction with respect to the surface. The scattering coefficient measures the ratio between the energy diffused in nonspecular directions and the total (specular and diffused) reflected energy. This parameter is useful when we are interested in modeling diffusion in reverberant enclosures but it does not account for the directions of the diffused wavefronts. This approximation is reasonable in the presence of a large number of diffusive reflections, but tends to become a bit restrictive when considering first-order diffusion only (i.e., ignoring diffusion of diffused paths). This is why in this paper we consider the additional assumption that diffusive surfaces be wide. This way the range of directions of diffused propagation turns out to be wide enough to minimize the impact of the above approximation. We will use the scattering coefficient to weight the contribution coming from totally diffuse reflections (modeled by Lambert's cosine law) and specular reflections.

3.2. Models of Diffraction. Diffraction is a very important propagation mode, particularly in densely occluded environments. Failing to properly account for this phenomenon in such situations could result in a poorly realistic rendering or even in annoying auditory artifacts. In this section we provide a brief description of three techniques for rendering diffraction phenomena: the Fresnel Ellipsoid, the line of sources, and the Uniform Theory of Diffraction (UTD). We will then explain why the UTD turns out to be the most suitable approach to the modeling of diffraction in conjunction with beam tracing.

3.2.1. Fresnel Ellipsoids. Let us consider a source S and a receiver R with an occluding obstacle in between. According to the Fresnel-Kirchhoff theory, the portion of the wavefront that is occluded by the obstacle does not contribute to the signal measured in R, which therefore differs from what we would have with unoccluded spherical propagation. In order to avoid using the Fresnel-Kirchhoff integral, we can adopt a simpler approach based on Fresnel ellipsoids. If d is the distance between S and R, only objects lying on paths whose length is between d and $d + \lambda/2$ are considered as obstacles, where λ is the wavelength. If x_s is the generic location of the secondary source, the locus of points that satisfy the equation $\overline{Sx_sR} - \overline{SR} \leq \lambda/2$ is an ellipsoid with foci in S and R. The portion of the ellipsoid that is occluded by obstacles provides an estimate of the absolute value of the diffraction filter's response. It is important to notice that the size of the Fresnel ellipsoid depends on the signal wavelength. As a consequence, in order to study diffraction in a given configuration, we need to estimate the occluded portion of the Fresnel ellipsoids at the frequencies of interest. In [26] the author proposes to use the graphics hardware to estimate the hidden portions of the ellipsoids. The main limit related to the Fresnel ellipsoid is the absence of information related to the phase of the signal: from the hidden portions of the ellipsoid, in fact, we can only infer the absolute value of the diffraction filter. If we need a more accurate rendering of diffraction, we must resort to other techniques.

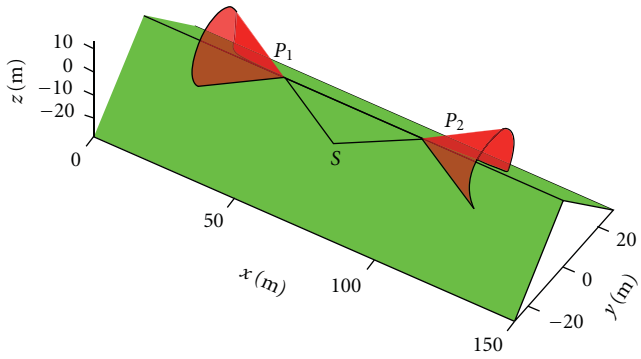


Figure 8: Geometric Theory of Diffraction: an acoustic source is in S. The acoustic source interacts with the obstacle, producing diffracted rays. Given the source position S, the points on the edge behave as secondary sources (e.g., P_1 and P_2 in the figure). According to the geometrical theory of diffraction, the angle between the outgoing rays and the edge equals the angle between the incoming ray and the edge. The envelope of the outgoing rays forms a cone, known in the literature as Keller cone.

3.2.2. Line of Sources. In [27] the authors propose a framework for accurately quantifying diffraction phenomena. Their approach is based on the fact that each point on a diffractive edge receives the incident ray and then re-emits a muffled version of it. The edge can therefore be seen as a line of secondary sources. The acoustic wave that reaches the receiver will then be a weighed superposition of all wavefronts produced by such edge sources.

In order to quantitatively determine the impact of diffraction in closed form, we need to be able to evaluate the visibility of a region (environment) from a line (edge of secondary sources). As far as we know, there are no results in the literature concerning the evaluation of regional visibility from a line. There are, however, several works that simplify the problem by sampling the line of sources. This way, visibility is evaluated from a finite number of points [28–30]. This last approach can be readily accommodated into our framework. However, as we are interested in a fast rendering of diffraction, we prefer to look into alternate formulations.

3.2.3. Uniform Theory of Diffraction. The Uniform Theory of Diffraction (UTD) was derived by Kouyoumjian and Pathak [15] from the Geometric Theory of Diffraction (GTD), proposed by Keller in 1962 [31]. As shown in Figure 8, according to the GTD, an acoustic ray that falls onto an edge with an angle θ_i produces a distribution of rays that lies on the surface of a cone. The axis of this cone is the edge itself, and its angle of aperture is $\theta_i = \theta_d$. The GTD assumes that the edge be of infinite extension, therefore, given a source and a receiver we can always find a point on the edge such that the diffracted path that passes through it will satisfy the constraint $\theta_i = \theta_d$. The Keller cones for the source S and two points P_1 and P_2 on the edge are shown in Figure 8.

In a way, the GTD allows us to compactly account for all contributions of a line distribution of sources. In fact, if we were to integrate all the infinitesimal contributions over an infinite edge, we would end up with only one significant path, which is the one that complies with the Keller condition, as all the other contributions would end up canceling each other out. The impact of diffraction on the source signal is rendered by a diffraction coefficient that depends on the frequency and on the angle between the incident ray and the angular aperture of the diffracting wedge (see Section 6.1 and [32] for further details). This geometric interpretation of diffraction is also adopted by the UTD. The difference between GTD and UTD is in how such diffraction coefficients are computed (see Section 6.1).

The use of the UTD in beam tracing is quite convenient as it only involves one incident ray per diffractive path. The UTD, however, assumes that the wedge be of infinite extension and perfectly reflective, which in some cases is too strong an assumption. Nonetheless, the advantages associated to considering only the shortest path make the UTD an ideal framework for accounting for diffraction in beam tracing applications. Notice that when the incident ray is orthogonal to the edge ($\theta_i = 90°$), the conic surface flattens onto a disc. This particular situation would be of special interest to us if we were considering an inherently 2D geometry. This, however, is NOT our case. We are, in fact, considering the situation of "separable" 3D environments [12], which result from the cartesian product between a 2D environment (floor map) and a 1D (vertical) direction. This special geometry (sort of an extruded floor map) requires the modeling of diffraction and diffusion phenomena in a 3D space. The Uniform Theory of Diffraction is, in fact, inherently three-dimensional, but our approach to the tracing of diffractive rays makes use of fast beam tracing, whose core is two dimensional. In order to be able to model UTD in fast beam tracing, we need therefore to first flatten the 3D geometry onto a 2D environment and later to adapt the 2D diffractive rays to the 3D nature of UTD. In order to clamp down the 3D geometry to the floor map, we need to establish a correspondence between the 3D geometric primitives that contribute to the Uniform Theory of Diffraction and some 2D geometric primitives. For example, when projected on a floor map, an infinitely long diffracting edge becomes a diffractive point, and a 3D diffracted ray becomes a 2D diffracted ray. When tracing diffractive beams, each wedge illuminated (directly or indirectly) by the source will originate a disk of diffracted rays, as shown in Figure 8. At this point we need to consider the 3D nature of the environment. We do so by "lifting" the diffracted rays in the vertical direction. We will end up with sort of an extruded cylinder containing all the rays that are diffracted by the edge. However, when we specify the locations of the source and the receiver, we find that this set includes also paths that do not honor the Keller cone condition $\theta_i = \theta_d$, and are therefore to be considered as unfeasible. The removal of all unfeasible diffracted rays can be done during the auralization phase. During the auralization, in fact, we select the paths coming from the closer diffractive wedges, as they are considered to be more perceptually relevant. The validation is a costly iterative process, therefore we only apply it to paths that are likely to be kept during the auralization.

3.3. New Needs and Requirements. As already said above, we are interested in extending the use of visibility maps for an accurate modeling and a fast rendering of diffusion and diffraction phenomena. As visibility diagrams were conceived for modeling specular reflections, it is important to discuss what needs and requirements need to be considered.

Diffraction. Visibility regions can be used for accommodating and modeling diffraction phenomena. In fact, according to the UTD, when illuminated by a beam, a diffractive edge becomes a virtual source with specific characteristics. Our goal is to model the indirect illumination of the receiver by means of secondary paths: wavefronts are emitted from the source, after an arbitrary number of reflections they fall onto the diffractive edge, which in turn illuminates the receiver after an arbitrary number of reflections. A common simplification that is adopted in works that deal with this phenomenon [14] consists of assuming that second and higher-order diffractions are of negligible impact onto the auralization result. This, in fact, is a perceptually reasonable choice that considerably reduces the complexity of the problem. In fact, a simple solution for implementing the phenomenon using the tracing tools at hand, consists of deriving a specialized beam-tree for each diffractive source. We will see later how. Another important aspect to consider in the modeling of diffraction is the Keller-cone condition [31], as briefly motivated above: with reference to Figure 8 we have to retain paths for which $\theta_i = \theta_d$. Tsingos et al. in [14] proposed to account for it by generating a reduced beam-tree, as constrained by a generalized cone that conservatively includes the Keller-cone. The excess rays that do not belong to the Keller cone, are removed afterwards through an appropriate check. We will see later that this approach can be implemented using the visibility diagrams.

Diffusion. Let us consider a source and a receiver, both facing a diffusive surface. In this case, each point of the surface generates an acoustic path between source and receiver. This means that the set of rays that emerge from the diffusing surface no longer form a beam (i.e., no virtual source can be defined as they do not meet in a specific point in space). In fact, according to Huygens principle, all points of the diffusive surface can be seen as secondary sources on a generally irregular surface, therefore we no longer have a single virtual source. Unlike diffraction, diffusion indeed poses new problems and challenges, as it prevents us from directly extending the beam tracing method in a straightforward fashion. One major difference from the specular case is the fact that the interaction between multiple diffusive surfaces cannot be described through an approach based on tracing, as we would have to face the presence of closed-loop diffusive paths. On the other hand, the impact of a diffusive surface on the acoustic field intensity is rather strong, therefore we cannot expect an acoustic path to still be of some significance after undergoing two or more diffusive reflections. It is thus quite reasonable to assume that any relevant acoustic paths would not include more than one relevant diffusive reflection along its way. We will see later on that this assumption, reasonably adopted by other authors as well (see [13]) opens the way to a viable solution to the real-time rendering of such acoustic phenomena. In fact, even if a diffusive surface does not preserve beam-like geometries, it is still possible to work on the visibility regions to speed up the tracing process between a source and a receiver through a diffusive reflection. This can be readily generalized to the case in which a chain of rays go from a source through a series of specular reflections and finally undergoes a diffusive reflection before reaching the receiver (diffusive path between a virtual source and a real receiver). A further generalization will be given for the case in which the rays undergo all specular reflections but one, which could be a diffusive reflection somewhere in between the chain. This last case corresponds to one diffusive path between a virtual source and a "virtual receiver", which can be computed by means of two intersecting beam-trees (a forward one from the source to the diffusive reflector and a backward one from the receiver to the diffusive reflector).

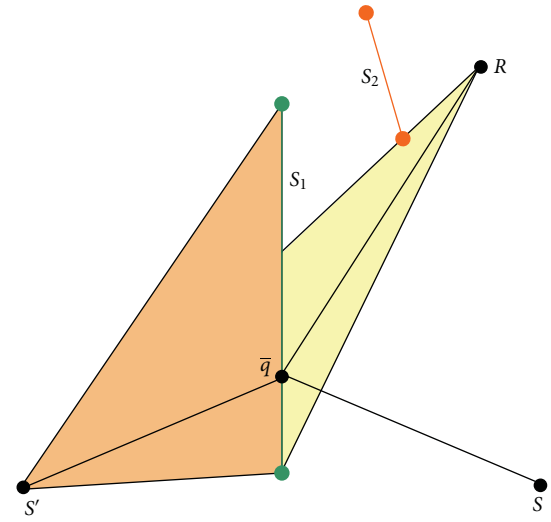


FIGURE 9: The real source S and the receiver R face the diffusive reflector s_1. Reflector s_2 partially occludes s_1 with respect to the receiver. S' is the image-source of S mirrored over the prolongation of s_1. The segments $\overline{S\overline{q}}$ and $\overline{\overline{q}R}$ form a diffusive path.

4. Tracing Diffusive Paths Using Visibility Diagrams

As already said before, the rendering of diffusion phenomena is commonly based on Bidirectional Beam Tracing, from both the source and the receiver. The need of tracing beams not just from the source but also from the receiver requires a certain degree of symmetrization in the definitions. For example, we need to introduce the concept of "virtual receiver", which is the location of the receiver as it gets iteratively mirrored against reflectors.

Let us consider the situation shown in Figure 9: a (virtual) source S and a (virtual) receiver R face the diffusive reflector s_1. Reflector s_2 partially occludes s_1 with respect

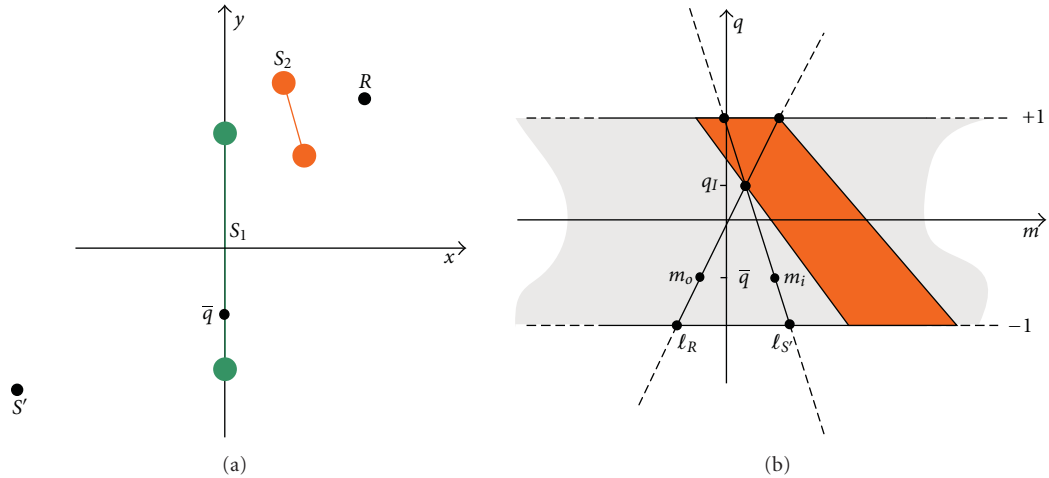


Figure 10: Normalized geometric domain (a) and corresponding dual space (b). The lines ℓ_R and $\ell_{S'}$ are the dual of the points S' and R.

to R. In order to simplify the problem we singled out the diffusive path $\overline{Sq}\ \overline{qR}$. Figure 9 also shows the image-source S', obtained by mirroring S over (the prolongation of) s_1. Notice that the geometry (lengths and angles) of the path $\overline{Sq}\ \overline{qR}$ is preserved if we consider the path $\overline{S'q}\ \overline{qR}$. We can therefore consider the virtual source S' instead of S, and trace the diffuse paths onto the visibility diagram "from" the reference diffusing reflector. If the coordinates of S' and R in the normalized geometric domain are $(x_{S'}, y_{S'})$ and (x_R, y_R), respectively, then the set of diffuse rays can be represented by

$$\{(\overline{m}_i, \overline{m}_o, \overline{q}) : y_{S'} = \overline{m}_i x_{S'} + \overline{q}, y_R = \overline{m}_o x_R + \overline{q}\}. \quad (3)$$

In other words, we are searching for the directions $\overline{m}_i$ and $\overline{m}_o$ of the rays that originate from the point $\overline{q}$ on the reference reflector and pass through S' and R.

The diffuse paths can be quite easily represented in the RRP from the reference reflector. The path from a point P on the reflector is, in fact, the intersection of the dual of P, which is the line $q = \overline{q}$; with the dual of S', which is the line $\ell_{S'} : q = -mx_{S'} + y_{S'}$. Similarly, the ray from P to R is the intersection between the line $q = \overline{q}$ and the line $\ell_R : q = -mx_R + y_R$. As we can see, we do not just have the ray that corresponds to the intersection of the two lines ℓ_R and $\ell_{S'}$ (same point, same direction), but a whole collection of rays corresponding to the horizontal segment that connects the source line $\ell_{S'}$ and the receiver line ℓ_R (same point but different directions).

Notice, however, that we have not yet considered potential occlusions of the diffuse paths from other reflectors in the environment. In Figure 9 we can see that only a portion of s_1 contributes to diffusion. In fact there is a portion of s_1 that is not visible from R, as it is occluded by s_2. This occlusion can be easily identified in the dual space by following a similar reasoning to that of (2) for the tracing of specular reflective paths. The set of rays that are potentially occluded by s_2 is represented in the dual space by the beam obtained by intersecting ℓ_R with the visibility region of s_2. In order to test whether the beam is actually occluded by s_2 or not, we can simply pick any ray within that area and check whether it reaches s_2 before R. The dual space representation of the problem of Figure 9 is described in the right-hand side of Figure 10 (the geometric description of the same problem is shown on the left-hand side of the same Figure, for reasons of convenience). In the example of Figure 10 the line of sight between R and s_1 is partially occluded by s_2, therefore only the segment $[-1, q_I]$ contributes to diffusive paths. Let us consider, for example, the path $\overline{S'q}\ \overline{qR}$, which is the line $q = q_P$. The directions of the rays from $\overline{q}$ to R and S' are given by m_i and m_o, respectively. Notice that until now, for reasons of simplicity, we have considered R to be the receiver, which forces the diffusion to occur last along the acoustic path. In order to make the proposed approach equivalent to bidirectional beam tracing, we will consider that R is the receiver or a virtual receiver obtained by building a beam-tree from the receiver location. In order to contain the computational cost, we only consider low-order virtual sources and virtual receivers. In Section 7, for example, we limit the order of virtual sources and virtual receivers to three.

5. Tracing Diffractive Beams and Paths Using Visibility Diagrams

In this section we extend the use of visibility diagrams to model diffractive paths and, using the UTD, we generalize the fast beam tracing method of [12] to account for this propagation phenomenon.

5.1. Selection of the Diffractive Wedges. As already discussed in [14], the diffractive field turns out to be less relevant when source and receiver are in direct visibility. The very first step of the algorithm consists, therefore, of selecting which edges are likely to generate a perceptually relevant diffraction. In what follows, we will refer to a *wedge* as a geometric configuration of two or more walls meeting into

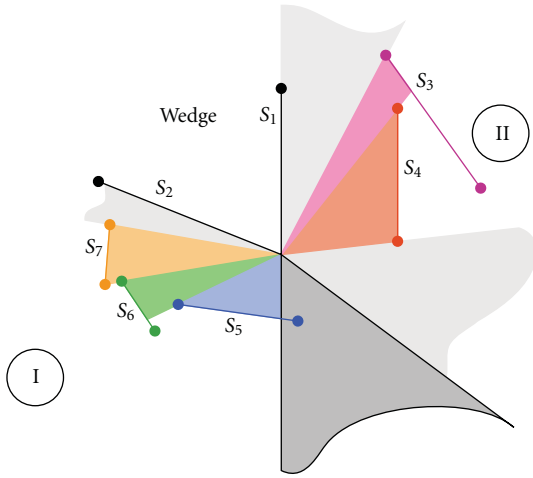


Figure 11: An example of diffractive wedge and beams departing from it. Notice that if either the source or the receiver fall outside the regions marked as I and II, then there is direct visibility, therefore diffraction can be neglected.

a single edge. If the angular opening of the wedge is smaller than π and both the receiver and the source fall inside the wedge, then source and receiver are in direct visibility. Not all wedges are, therefore, worth retaining. Even if a wedge is diffractive, we can still find configurations where source and receiver are in direct visibility. When this happens, diffraction is less relevant than the direct path and we discard these diffractive paths. With reference to Figure 11, we are interested in auralizing diffraction in the two regions marked as I and II, where source and receiver are not necessarily in conditions of mutual visibility. For each of the two regions we will build a beam-tree. This selection process returns a list $i = 1, \ldots, M$ of diffractive wedges and their coordinates.

5.2. Tracing Diffractive Beam-Trees. With reference to Figure 12, our goal is to split the bundle of rays departing from the virtual source and directed towards the regions I and II into subbeams. In order to do this, we take advantage of the visibility diagrams. As seen in Section 3, the virtual source is placed on the tip of the diffracting wedge. The dual of this virtual source is the semi-infinite line of equation $q = +1$ or $q = -1$, the sign depending on the normalization of the RRP. We are interested in auralizing the diffracted field only in regions I and II, therefore we evaluate the regional visibility along the lines (virtual sources in the geometric space) $q = \pm 1$ only for $m > m_0$ or $m < m_0$. By scanning these semi-infinite lines along the visibility diagrams we can immediately determine all the visible reflectors of the above beams, and therefore determine their branching. Figure 12 shows the first-level beams and their dual representations for the configuration of Figure 11 (which refers to regions I and II). The propagation and the branching of these diffractive beams will follow the usual rules, according to the iterative tracing mechanism defined in Section 2 for specular reflections. The number of levels (branching order) of the diffractive beam-trees is indeed to be specified in a different fashion compared with reflective beam-trees, for reasons of relevance and computational load. We notice that, at this stage, the location of source or receiver does not need to be specified. As a consequence, we can build diffractive beam-trees in a precomputation phase.

5.3. Diffractive Paths Computation. Once source and receiver are specified, we can finally build the diffractive paths. Let $\mathcal{P}_{S,\mathrm{I}}^{(i,m)}(\mathcal{P}_{S,\mathrm{II}}^{(i,m)})$ denote the ith reflective path between the source and the beam-tree to the region I (to the region II) departing from the diffractive wedge marked with m in the environment. As far as the auralization of diffraction is concerned, the path $\mathcal{P}_{S,\mathrm{I}}^{(i,m)}$ is completely defined if we specify the source location, the position of the point of incidence of the ray on the walls (possibly in normalized coordinates) and the location of the diffractive edge. The set of paths between the source and the diffractive region inside the beam-tree of the region I (II) is $\mathcal{P}_{S,\mathrm{I}}^{(m)}$ $(\mathcal{P}_{S,\mathrm{I}}^{(m)})$. Similarly, $\mathcal{P}_{R,\mathrm{I}}^{(j,n)}$ $(\mathcal{P}_{R,\mathrm{I}}^{(j,n)})$ is the jth path between the receiver and the nth diffractive wedge in the beam-tree I (II), and $\mathcal{P}_{R,\mathrm{I}}^{(n)}$ $(\mathcal{P}_{R,\mathrm{II}}^{(n)})$ is the set of paths between receiver and the diffractive edge inside the region I (II).

We are interested in diffractive paths where source and receiver fall in opposite regions: if the source falls within the beam-tree I, then we will check whether the receiver is in the beam-tree II, as diffraction is negligible when both source and receiver are in the same region. Let $\mathcal{P}_{\mathrm{I}\to\mathrm{II}}^{(m)}$ be the set of paths associated to the diffractive edge m (when the source is in beam-tree I, and the receiver in II). Similarly, $\mathcal{P}_{\mathrm{II}\to\mathrm{I}}^{(m)}$ is the set of paths where the source is in the beam-tree II and the receiver is in I. These sets are obtained as

$$\begin{aligned}\mathcal{P}_{\mathrm{I}\to\mathrm{II}}^{(m)} &= \mathcal{P}_{S,\mathrm{I}}^{(m)} \otimes \mathcal{P}_{R,\mathrm{II}}^{(m)},\\ \mathcal{P}_{\mathrm{II}\to\mathrm{I}}^{(m)} &= \mathcal{P}_{S,\mathrm{II}}^{(m)} \otimes \mathcal{P}_{R,\mathrm{I}}^{(m)},\end{aligned} \tag{4}$$

where $\otimes$ denotes the cartesian product. Finally, the set of paths for the diffractive edge m is the union of $\mathcal{P}_{\mathrm{I}\to\mathrm{II}}^{(m)}$ and $\mathcal{P}_{\mathrm{II}\to\mathrm{I}}^{(m)}$:

$$\mathcal{P}^{(m)} = \mathcal{P}_{\mathrm{I}\to\mathrm{II}}^{(m)} \cup \mathcal{P}_{\mathrm{II}\to\mathrm{I}}^{(m)}. \tag{5}$$

In order to preserve the Keller cone condition, we have to determine the point P_d on the diffractive edge that makes the angle θ_i between the incoming ray and the edge equal to the angular aperture of the Keller cone θ_d. When dealing with diffractive beam-trees which include also one or more reflections, the condition $\theta_i = \theta_d$ is no longer sufficient for determining the location of the virtual source, as we must also preserve the Snell's law for the reflections inside the path. In [14] the authors propose to compute the diffraction and reflection points along the diffractive path through a system of nonlinear equations. This solution is obtained through an iterative Newton-Raphson algorithm. In this paper we adopt the same approach.

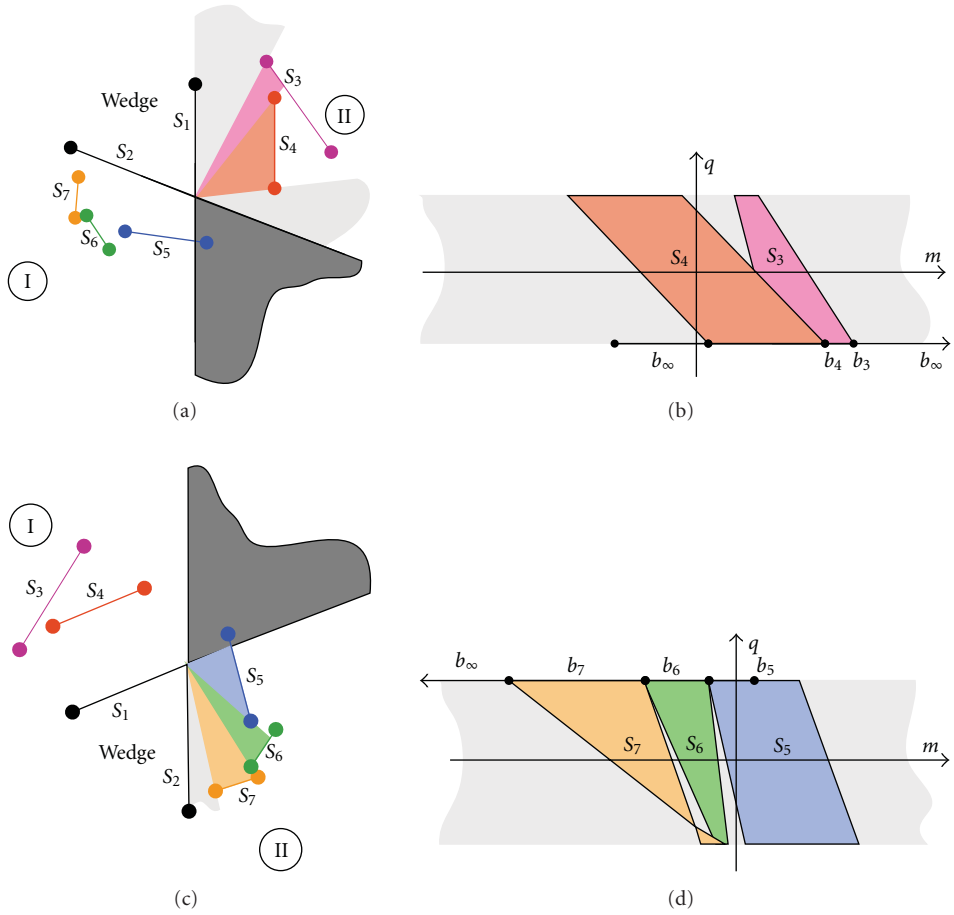


FIGURE 12: When the source and the receiver fall outside the regions I and II, diffraction becomes negligible. The diffractive beam-trees will therefore cover regions I and II only.

6. Applications to Rendering

In this section we discuss the above beam tracing approach in the context of acoustic rendering. As anticipated, we first discuss the more traditional case of channel-based (point-to-point) rendering. This is the case of auralization when the user is wearing a headset.

We will then discuss the case of geometric wavefield rendering.

6.1. Path-Tracing for Channel Rendering. In this section we propose and describe a simple auralization system based on the solutions proposed above. Due to their different nature, we will distinguish between reflective, diffusive and diffractive echoes. In particular, diffraction involves a perceptually relevant low-pass filtering on the signal, hence we will stress diffraction instead of diffusion and reflection. The diffraction is rendered by a coefficient, whose value depends on the frequency. Each diffractive wedge, therefore, acts like a filter on the incoming signal, with an apparent impact on the overall computational cost. We therefore made our auralization algorithms select the most significant paths only, based on a set of heuristic rules that take the power of each diffractive filter into account.

6.1.1. Auralization of Reflective Paths. As far as reflections are concerned, we follow the approach proposed in [12]: the echo i is characterized by a length ℓ_i. The magnitude of the ith echo is $A_i = r^k/\ell_i$, r being the reflection coefficient and k being the number of reflections (easily determined by inspecting the beam-tree). The delay associated to the echo i is $d_i = \ell_i/c$, where c is the speed of sound ($c = 340$ m/s in our experiments).

6.1.2. Auralization of Diffraction. As motivated in Section 5, we have resorted to UTD to render diffraction. In order to auralize diffraction paths, we have to compute a diffraction coefficient, which exhibits a frequency-dependent behavior. A comprehensive tutorial on the computation of the diffraction coefficient in UTD may be found in [14]. We remark here that in order to compute the diffractive filter, for each path we need some geometrical information about the wedge (available at a precomputation stage) and about the path, which is available only after the Newton Raphson algorithm described in the previous Section.

6.1.3. Auralization of Diffusive Paths. Accurate modeling of diffusion is typically based on statistical methods [33]. Our

modeling solution is, in fact, aimed at auralization and rendering applications, therefore we resort to an approximated but still reliable approach that does not significantly impact on the computational cost: we make use of the Lambert cosine law, which works for energies, to compute the energy response. Then we convert the energy response onto a pressure response by taking its square root. A similar idea was developed in [34], where the author combines beam tracing and radiosity.

In order to compute the energy response, we need the knowledge of the position of the virtual source and receiver and the segment on the illuminating reflector from which rays depart. The energy response $H_d^{(i)}$ of the diffusive path from reflector i is given by the integration of the contributions of each point of the reflector. If the length of the diffusive portion is ℓ, then the auralization filter may be approximated as

$$H_d^{(i)} = \sum_{i=1}^{N} D(P_i)\Delta P, \tag{6}$$

where P_i is the coordinate on the diffuser, N is the number of portions in which the segment has been subdivided and $D(P_i)$ is modeled according to the Lambert cosine law:

$$D(P_i) = \frac{\cos[\theta_i(P_i)]\cos[\theta_o(P_i)]}{\pi}, \tag{7}$$

and $\theta_i(P_i)$ and $\theta_o(P_i)$ are the directions of the incident and outgoing rays referred to the axis of the reflector, respectively. Angles $\theta_i(P_i)$ and $\theta_o(P_i)$ are related to m_i and m_o according to

$$\begin{aligned} \theta_i(P) &= \tan[m_i(P)], \\ \theta_o(P) &= \tan[m_o(P)]. \end{aligned} \tag{8}$$

6.2. Beam Tracing for Sound Field Rendering. The beam-tree contains all the information that we need to structure a sound field as a superposition of individual beams, each originating from a different image-source. A solution was recently proposed for reconstructing an individual beam in a parametric fashion using a loudspeaker array [16]. Here, an arbitrary beam of prescribed aperture, orientation origin is reconstructed using an array of loudspeakers. In particular, the least squares difference of the wavefields produced by the array of loudspeakers and by the virtual source is minimized over a set of predefined control points. The minimization returns a spatial filter to be applied at each loudspeaker. It is interesting to notice that the approach described in [16] offers the possibility to design a spatial filter that performs a nonuniform weighting of the rays within the same beam. This feature enables the rendering of "tapered" beams, which is particularly useful when dealing with diffractive beams. In fact, the diffraction coefficient (see [14] for further details) assigns different levels of energy to rays within the same beam, according to the reciprocal geometric configuration of the source, the wedge and the direction of travel of the ray departing from the wedge.

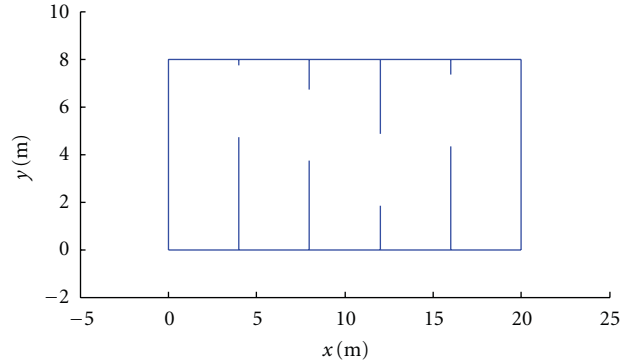


Figure 13: Example of a test environment used for assessing the computational efficiency of our beam tracing algorithm. The 8 m × 4 m rooms are all connected together through randomly-located apertures.

The rendering of the overall sound field is finally achieved by adding together the spatial filters for the individual beams. More details and a some preliminary results of this method can be found in [35].

7. Experimental Results

In order to test the validity of our solution, we performed a series of simulation experiments as well as a measurement campaign in a real environment. An initial set of simulations was performed with the goal of assessing the computational efficiency of our techniques for the auralization of reflective, diffractive and diffusive paths, separately considered. In order to assess the accuracy of our method, were constructed the impulse responses of a given environment (on an assigned grid of points) through both simulation and direct measurements. From such impulse responses, we derived a set of parameters, typically used for describing reverberation. The comparison between the measured parameters and the simulated ones was aimed at assessing the extent of the improvement brought by introducing cumulatively diffraction and diffusion into our simulation.

7.1. Computation Time. In order to assess the computational efficiency of the proposed method, we conducted a series of simulations in environments of controlled complexity (variable geometry). We developed a procedure for generating testing environments with an arbitrary number of 8 m × 4 m rooms, all chained together as shown in Figure 13. An N-room environment is therefore made of $4N$ reflectors. In order to enable acoustic interaction between rooms, each one of the intermediate walls exhibits a randomly-placed opening. In our tests, we generated environments with 20, 40, 80, 120, 320, 640 such walls. The simulation platform was based on an Intel Mobile Pentium processor equipped with 1 GB of RAM, and the goal was to:

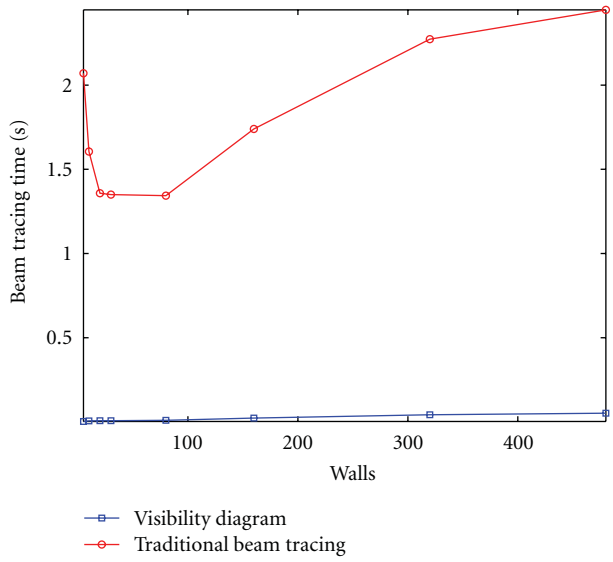


Figure 14: Beam-tree building time for variable geometry for traditional beam tracing (circles) and visibility-based beam tracing (squares) for 10,000 beams. The proposed approach greatly outperforms traditional beam tracing especially when the number of traced beams is very large.

(i) compare the beam-tree's building time of visibility-based beam tracing with that of traditional beam tracing [5];

(ii) measure the diffractive path-tracing time;

(iii) measure the diffusive path-tracing time.

As far as the time spent by the system in computing the visibility diagrams is concerned, we invite the reader to [12]. The beam-tree building time is intended as the time spent by the algorithm in tracing a preassigned number of beams. In order to assess the impact of the proposed approach on this specific parameter, in comparison with a traditional beam-tracing approach, we stopped the algorithm when 10,000 beams were traced. The simulation results are shown in Figure 14. As expected, our approach turns out to outperform traditional beam tracing. As the beam-tree building time strongly depends on the source location, we conducted numerous tests by placing the source at the center of each one of the rooms of the modular environments. The beam-tree building time was then computed as the average of the beam-tree building times over all such simulations. In order to test the tracing time of the diffusive paths the source has been placed in the center the modular environment. We conducted several experiments placing the receiver at the center of each one of the rooms. We measured the time spent by the system in computing the diffusive paths. We considered up to three reflections before and after the diffuse interaction. Figure 15 shows the average tracing time of the diffusive paths as a function of the number of walls. If we are interested in an accurate rendering, we should trace at least 1000 beams: in this situation we notice that even in a complex environment (e.g., 640 walls) the auralization of the diffusive paths takes only a fraction of the time spent by the

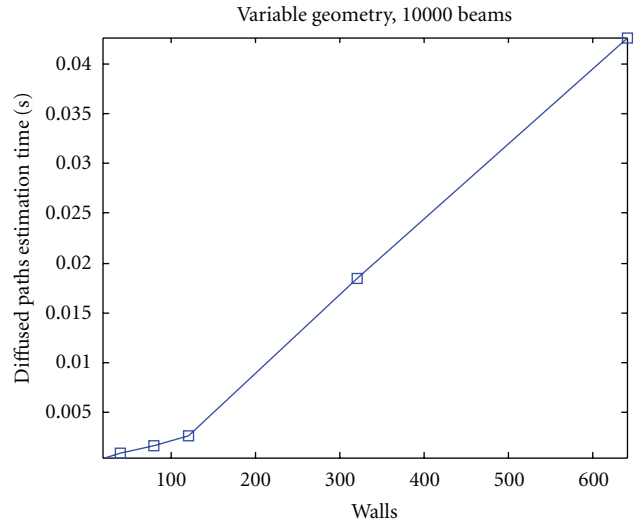


Figure 15: Tracing time of the diffusive paths as a function of the number of walls of the environment of Figure 13.

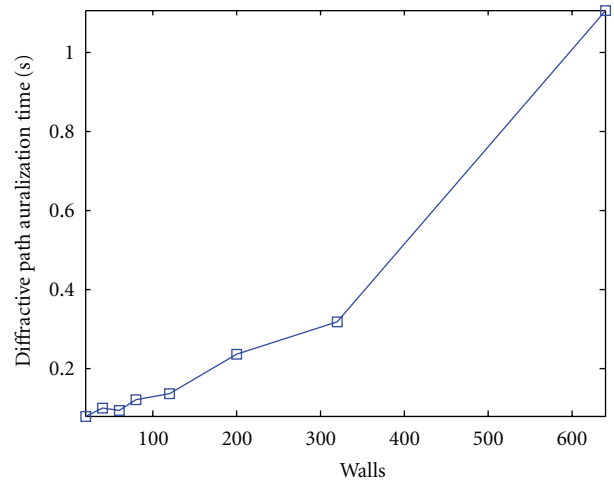


Figure 16: Auralization time for diffractive paths (measured in seconds) versus the complexity (measured by the number of walls) of the environment.

system for the auralization of the reflective paths. The last test is aimed at assessing the computational time for auralizing the diffractive paths. As done above, the source was placed at the center of the environment and we conducted several experiments, placing the receiver at the center of each room. In particular in this situation two tests have been conducted:

(i) the number of diffractive paths with respect to the number of walls in the environment;

(ii) the computational time for auralizing the diffractive paths.

Figure 16 shows the time that the system takes to auralize the diffractive paths, as a function of the number of walls in the environment. Figure 17 shows the number of paths for the same experiment. The number of diffractive paths depends upon both the depth of the diffractive beam-trees

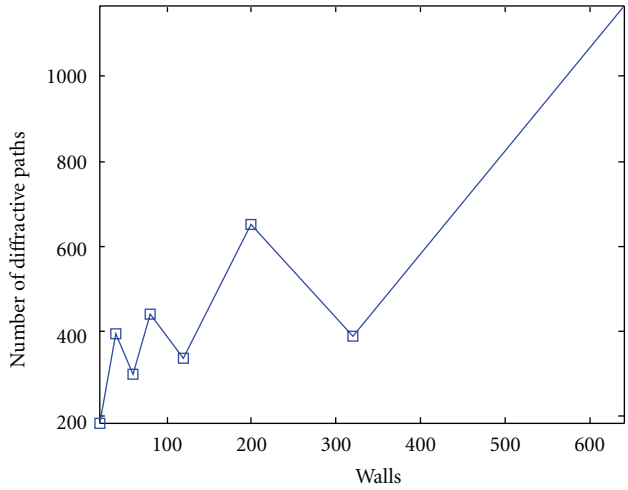


Figure 17: Number of diffractive paths versus the complexity of the environment (measured by the number of walls) in which experiments are done.

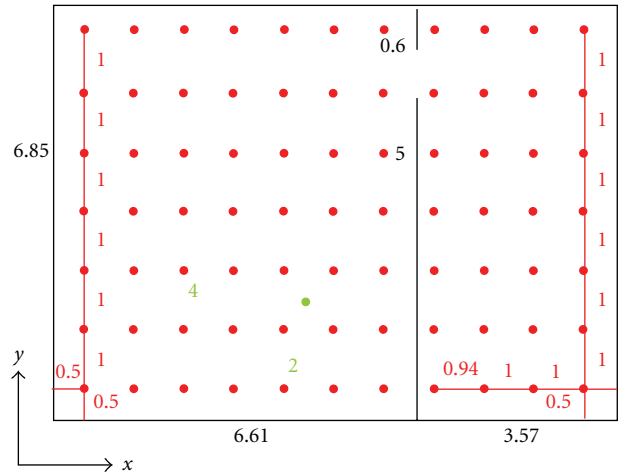


Figure 18: Floor-map of the validation environment: the grid represents the 77 acquisition points and the loudspeaker location is marked with a dot.

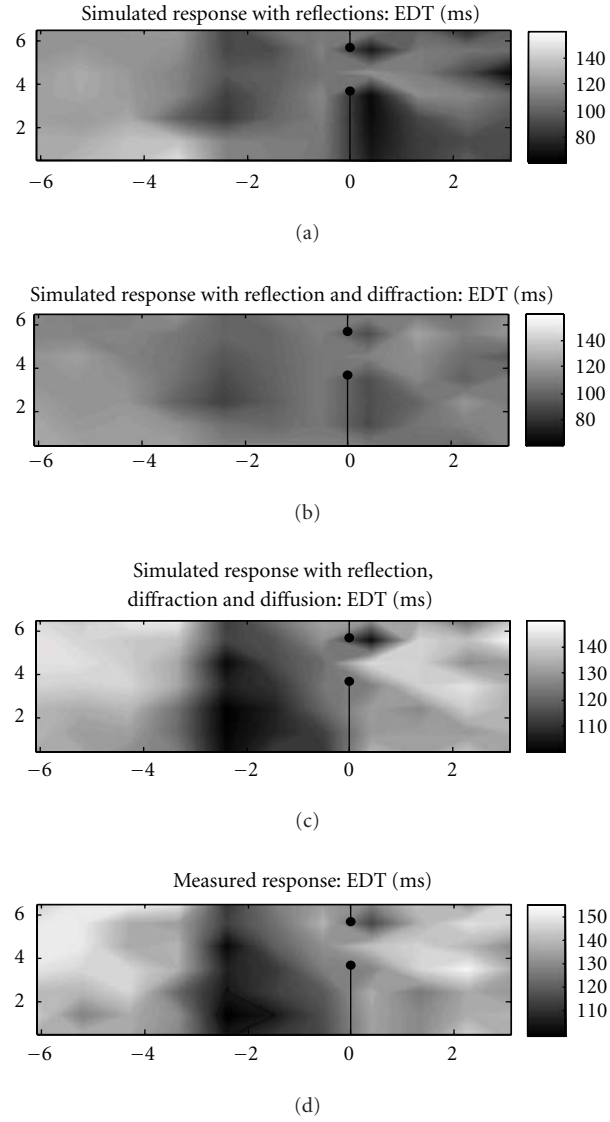


Figure 19: Early Decay Time map: (a) simulated with reflections only, (b) simulated with reflections and diffraction, (c) simulated with reflections, diffraction and diffusion, (d) measured.

and the number of diffractive edges in the environment in quite an unpredictable fashion. Notice that the auralization for the diffractive paths can exceed the auralization of the reflective paths. However, we should keep in mind that the computation of over 1000 diffractive paths can be quite redundant from a perceptual standpoint. We propose here to select the diffractive paths arisen from the closer diffractive wedges, as they are more perceptually relevant.

7.2. Validation. It becomes important to assess how well spatial distributions of reverberations are rendered by this approach. In order to do so, we conducted comparative tests between simulated data and real data with corresponding geometries. We placed a high-quality loudspeaker in a reverberant environment and used it to reproduce a MLS sequence [36]. The environment's floor map is shown in Figure 18, where the location of the loudspeaker and microphone are marked with green and red dots, respectively. The acquired signals were then processed to determine the impulse responses of the environment in those locations, as described in [36]. The reflective and diffusive reflection coefficients of the walls, indeed, could not be directly measured but were coarsely determined so that the T_{20} of the simulated impulse response would approximate the estimated one in a given test position. As a sample-to-sample comparison of the acquired and simulated impulse responses is not very informative, we compared some parameters that describe the impulse response, which can be readily derived from the simulated and measured impulse responses. In particular, we run extensive simulations first with reflections only (up to the 16th order of the beam-tree), with diffraction (up to the 6th order) and finally with diffusion (with an order 3).

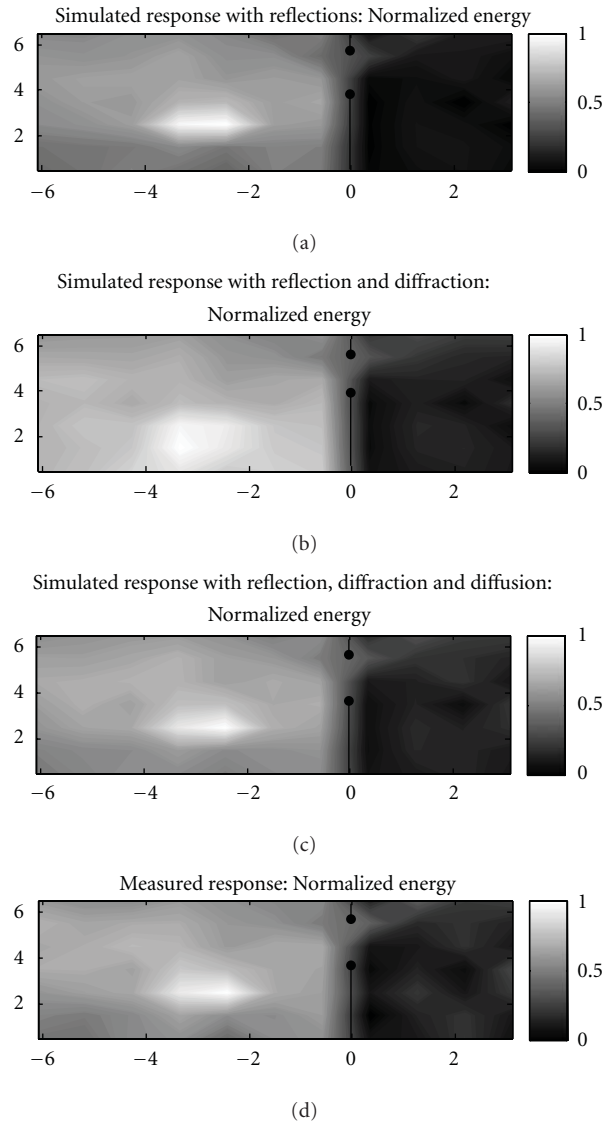


Figure 20: Normalized energy map: (a) simulated with reflections only, (b) simulated with reflections and diffraction, (c) simulated with reflections, diffraction and diffusion, (d) measured.

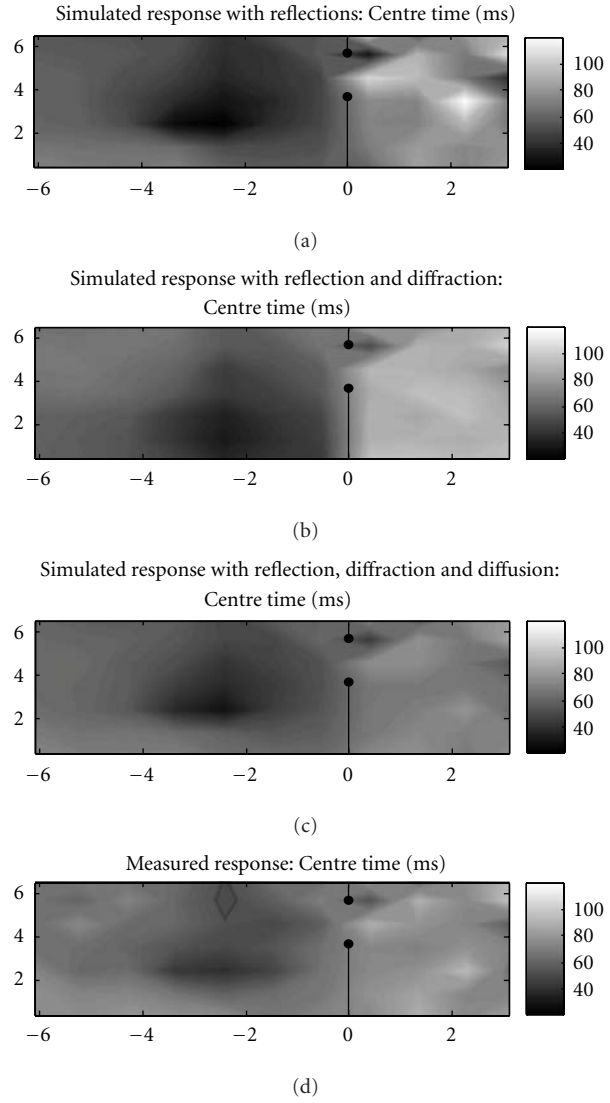


Figure 21: Center Time map: measured (a) simulated with reflections only, (b) simulated with reflections and diffraction, (c) simulated with reflections, diffraction and diffusion, (d) measured.

Early Decay Time (EDT). EDT is the time that the Schroeder envelope of the impulse response takes to drop of 10 dB. A comparison of the EDT maps (geographical distribution of the EDT in the measured and simulated cases) is shown in Figure 19. Notice that the impact of diffusion in this experiment is quite significant: in fact, as expected, including diffusion increases the energy in the "tail" of the impulse response, with the result of obtaining a more realistic reverberation.

Normalized Energy of the Impulse Response. where the normalization is performed with respect to the impulse response of maximum energy. A comparison between normalized energy maps is shown in Figure 20. Once again we notice that a simulation of all phenomena (reflections, diffraction and diffusion) produces more realistic results.

Center Time. It is first-order momentum of the squared impulse response. The Center Time (CT) map is shown in Figure 21, where we see that a simulation based on reflections, diffraction and diffusion allows us to achieve, once again, a good match. We notice, in particular, that modeling diffusion increases the level of smoothness of the simulated map, which makes it more similar to the measured one.

8. Conclusions

In this paper we proposed an extension of the visibility-based beam tracing method proposed in [12], which now allows us to model and render propagation phenomena such as diffraction and diffusion, without significantly affecting the computational efficiency. We also improved the method

in [12] by showing that not just the construction of the beam-tree but also the whole path-tracing process can be entirely performed on the visibility maps. We finally showed that this approach produces quite accurate results when comparing simulated data with real acquisitions. Thanks to that, this modeling tool proves particularly useful every time there is a need for an accurate and fast simulation of acoustic propagation in environments of variable geometry and variable physical characteristics.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 226007.

References

[1] S. Kopuz and N. Lalor, "Analysis of interior acoustic fields using the finite element method and the boundary element method," *Applied Acoustics*, vol. 45, no. 3, pp. 193–210, 1995.

[2] A. Kludszuweit, "Time iterative boundary element method (TIBEM)—a new numerical method of four-dimensional system analysis for thecalculation of the spatial impulse response," *Acustica*, vol. 75, pp. 17–27, 1991.

[3] R. Ciskowski and C. Brebbia, *Boundary Element Methods in Acoustics*, Elsevier, Amsterdam, The Netherlands, 1991.

[4] U. Krockstadt, "Calculating the acoustical room response by the use ofa ray tracing technique," *Journal of Sound and Vibrations*, vol. 8, no. 1, pp. 118–125, 1968.

[5] Paul S. Heckbert and Hanrahan, "Beam tracing polygonal objects," in *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 119–127, July 1984.

[6] N. Dadoun, D. Kirkpatrick, and J. Walsh, "The geometry of beam tracing," in *Proceedings of the ACM Symposium on Computational Geometry*, pp. 55–61, Sedona, Ariz, USA, June 1985.

[7] M. Monks, B. Oh, and J. Dorsey, "Acoustic simulation and visualisationusing a new unified beam tracing and image source approach," in *Proceedings of the 100th Audio Engineering Society Convention (AES '96)*, Los Angeles, Ariz, USA, 1996.

[8] U. Stephenson and U. Kristiansen, "Pyramidal beam tracing and time dependent radiosity," in *Proceedings of the 15th International Congress on Acoustics*, pp. 657–660, Trondheim, Norway, June 1995.

[9] J. P. Walsh and N. Dadoun, "What are we waiting for? The developmentof Godot, II," *Journal of the Acoustical Society of America*, vol. 71, no. S1, p. S5, 1982.

[10] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West, "Beam tracing approach to acoustic modeling for interactive virtual environments," in *Proceedings of the Annual Conference on Computer Graphics (SIGGRAPH '98)*, pp. 21–32, July 1998.

[11] S. Laine, S. Siltanen, T. Lokki, and L. Savioja, "Accelerated beam tracing algorithm," *Applied Acoustics*, vol. 70, no. 1, pp. 172–181, 2009.

[12] F. Antonacci, M. Foco, A. Sarti, and S. Tubaro, "Fast tracing of acoustic beams and paths through visibility lookup," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 4, pp. 812–824, 2008.

[13] T. Funkhouser, P. Min, and I. Carlbom, "Real-time acoustic modeling for distributed virtual environments," in *Proceedings of ACM Computer Graphics (SIGGRAPH '99)*, A. Rockwood, Ed., pp. 365–374, Los Angeles, Ariz, USA, 1999.

[14] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom, "Modeling acoustics in virtual environments using the uniform theory of diffraction," in *Proceedings of the 28th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001)*, pp. 545–552, August 2001.

[15] R. G. Kouyoumjian and P. H. Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface," *Proceedings of the IEEE*, vol. 62, no. 11, pp. 1448–1461, 1974.

[16] A. Canclini, A. Galbiati, A. Calatroni, F. Antonacci, A. Sarti, and S. Tubaro, "Rendering of an acoustic beam through an array of loud-speakers," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx '09)*, 2009.

[17] A. J. Berkhout, "Holographic approach to acoustic control," *Journal of the Audio Engineering Society*, vol. 36, no. 12, pp. 977–995, 1988.

[18] M. Foco, P. Polotti, A. Sarti, and S. Tubaro, "Sound spatialization basedon fast beam tracing in the dual space," in *Proceedings of the 6th International Conference on Digital Audio Effects (DAFX '03)*, pp. 198–202, London, UK, September 2003.

[19] B. Brouard, D. Lafarge, J. -F. Allard, and M. Tamura, "Measurement and prediction of the reflection coefficient of porous layers at oblique incidence and for inhomogeneous waves," *Journal of the Acoustical Society of America*, vol. 99, no. 1, pp. 100–107, 1996.

[20] M. Kleiner, H. Gustafsson, and J. Backman, "Measurement of directional scattering coefficients using near-field acoustic holography and spatial transformation of sound fields," *Journal of the Audio Engineering Society*, vol. 45, no. 5, pp. 331–346, 1997.

[21] C. Nocke, "In-situ acoustic impedance measurement using a free-field transfer function method," *Applied Acoustics*, vol. 59, no. 3, pp. 253–264, 2000.

[22] S. I. Thomasson, "Reflection of waves from a point source by an impedance boundary," *Journal of the Acoustical Society of America*, vol. 59, no. 4, pp. 780–785, 1976.

[23] T. Funkhouser, N. Tsingos, and J. M. Jot, "Sounds good to me! computational sound for graphics, virtual reality, and interactive systems," in *Proceedings of ACM Computer Graphics (SIGGRAPH '02)*, San Antonio, Tex, USA, July 2002.

[24] H. Kuttruff, *Room Acoustics*, Elsevier, Amsterdam, The Netherlands, 3rd edition, 1991.

[25] L. L. Beranek, *Concert and Opera Halls: How they Sound*, Acoustical Society of America through the American Institute of Physics, 1996.

[26] N. Tsingos, *Simulating high quality virtual sound fields for interactive graphics applications*, Ph.D. dissertation, Universite J. Fourier, Grenoble, France, 1998.

[27] M. A. Biot and I. Tolstoy, "Formulation of wave propagation in infinite media by normal coordinates with an application to diffraction," *Journal of the Acoustical Society of America*, vol. 29, pp. 381–391, 1957.

[28] T. Lokki, L. Savioja, and P. Svensson, "An efficient auralization of edge diffraction," in *Proceedings of 21st International Conference of Audio Engineering Society (AES '02)*, May 2002.
[29] U. P. Svensson, R. I. Fred, and J. Vanderkooy, "An analytic secondary source model of edge diffraction impulse responses," *Journal of the Acoustical Society of America*, vol. 106, no. 5, pp. 2331–2344, 1999.
[30] R. R. Torres, U. P. Svensson, and M. Kleiner, "Computation of edge diffraction for more accurate room acoustics auralization," *Journal of the Acoustical Society of America*, vol. 109, no. 2, pp. 600–610, 2001.
[31] J. B. Keller, "Geometrical theory of diffraction," *Journal of the Optical Society of America*, vol. 52, pp. 116–130, 1962.
[32] D. McNamara, C. Pistorius, and J. Malherbe, *Introduction to the Uniform Geometrical Theory of Diffraction*, Artech House, 1990.
[33] J. H. Rindel, "The use of computer modeling in room acoustics," *Journal of Vibroengineering*, vol. 3, pp. 41–72, 2000.
[34] T. Lewers, "A combined beam tracing and radiatn exchange computer model of room acoustics," *Applied Acoustics*, vol. 38, no. 2–4, pp. 161–178, 1993.
[35] F. Antonacci, A. Calatroni, A. Canclini, A. Galbiati, A. Sarti, and S. Tubaro, "Soundfield rendering with loudspeaker arrays through multiple beam shaping," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '09)*, pp. 313–316, 2009.
[36] D. D. Rife and J. Vanderkooy, "Transfer-function measurement with maximum-length sequences," *Journal of the Audio Engineering Society*, vol. 37, no. 6, pp. 419–444, 1989.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 719197, 11 pages
doi:10.1155/2010/719197

Research Article

A Stereo Crosstalk Cancellation System Based on the Common-Acoustical Pole/Zero Model

Lin Wang,[1,2] Fuliang Yin,[1] and Zhe Chen[1]

[1] *School of Electronic and Information Engineering, Dalian University of Technology, Dalian 116023, China*
[2] *Institute for Microstructural Sciences, National Research Council Canada, Ottawa, ON, Canada K1A 0R6*

Correspondence should be addressed to Lin Wang, wanglin_2k@sina.com

Received 8 January 2010; Revised 21 June 2010; Accepted 7 August 2010

Academic Editor: Augusto Sarti

Crosstalk cancellation plays an important role in displaying binaural signals with loudspeakers. It aims to reproduce binaural signals at a listener's ears via inverting acoustic transfer paths. The crosstalk cancellation filter should be updated in real time according to the head position. This demands high computational efficiency for a crosstalk cancellation algorithm. To reduce the computational cost, this paper proposes a stereo crosstalk cancellation system based on common-acoustical pole/zero (CAPZ) models. Because CAPZ models share one set of common poles and process their zeros individually, the computational complexity of crosstalk cancellation is cut down dramatically. In the proposed method, the acoustic transfer paths from loudspeakers to ears are approximated with CAPZ models, then the crosstalk cancellation filter is designed based on the CAPZ transfer functions. Simulation results demonstrate that, compared to conventional methods, the proposed method can reduce computational cost with comparable crosstalk cancellation performance.

1. Introduction

A 3D audio system can be used to position sounds around a listener so that the sounds are perceived to come from arbitrary points in space [1, 2]. This is not possible with classical stereo systems. Thus, 3D audio has the potential of increasing the sense of realism in music or movies. It can be of great benefit in virtual reality, augmented reality, remote video conference, or home entertainment. A 3D audio technique achieves virtual sound perception by synthesizing a pair of binaural signals from a monaural source signal with the provided 3D acoustic information: the distance and direction of the sound source with respect to the listener. Specifically, the sense of direction can be rendered by using head-related acoustic information, such as head-related transfer functions (HRTFs) which can be obtained by either experimental or theoretical means [3, 4]. To deliver binaural signals, the simplest way is through headphones. However, in many applications, for example, home entertainment environment, teleconferencing, and so forth, many listeners prefer not to wear headphones. If loudspeakers are used, the delivery of these binaural signals to the listener's ears is not straightforward. Each ear receives a so-called crosstalk component, moreover, the direct signals are distorted by room reverberation. To overcome the above problems, an inverse filter is required before playing binaural signals through loudspeakers.

The concept of crosstalk cancellation and equalization was introduced by Atal and schroeder [5] and Bauer [6] in the early 1960s. Many sophisticated crosstalk cancellation algorithms have been presented since then, using two or more loudspeakers for rendering binaural signals. Crosstalk cancellation can be realized directly or adaptively. Supposing that the acoustical transfer paths from loudspeakers to ears are known, the direct implementation method calculates the crosstalk cancellation filter by directly inverting the acoustical transfer functions [7, 8]. Generally a head-tracking scheme, which can tell the head position precisely, is employed to work together with the direct estimation method. The direct estimation method can be implemented in the time or frequency domain. Time-domain algorithms are generally computationally consuming, while frequency-domain algorithms have lower complexity. On the other hand, time-domain algorithms perform better than

frequency-domain ones with the same crosstalk cancellation filter length. For example, a frequency-domain method such as the fast deconvolution method [7], which has been shown to be very useful and easy to use in several practical cases, can suffer from a circular convolution effect when the inverse filters are not long enough compared to the duration of the acoustic path response. In an adaptive implementation method, the crosstalk cancellation filter is calculated adaptively with the feedback signals received by miniature microphones placed in human ears [9]. Several adaptive crosstalk cancellation methods typically employ some variation of LMS or RLS algorithms [10–13]. The LMS algorithm, which is known for its simplicity and robustness, has been used widely, but its convergence speed is slow. The RLS algorithm may accelerate the convergence, but the large computation load is a side effect. Although many algorithms have been proposed, the adaptive implementation method remains academic research rather than a real solution. The reason is that people who do not want to use headphones would probably not like to use a pair of microphones in the ears to optimize loudspeaker reproduction either.

One key limitation of a crosstalk cancellation system arises from the fact that any listener movement which exceeds 75–100 mm may completely destroy the desired spatial effect [14, 15]. This problem can be resolved by tracking the listener's head in 3D space. The head position is captured by a magnetic or camera-based tracker, then the HRTF filters and the crosstalk canceller based on the location of the listener are updated in real time [16]. Although head-tracking systems can be employed, measures should still be taken to increase the robustness of the crosstalk cancellation system. It has been shown that the robust solution to this virtual sound system could be obtained by placing the loudspeakers in an appropriate way to ensure that the acoustic transmission path or transfer function matrix is well conditioned [17–19]. Robust crosstalk cancellation methods with multiple loudspeakers have been proposed [8, 20, 21]. Another approach adds robustness of a crosstalk canceller by exploring the statistical knowledge of acoustic transfer functions [22].

This paper focuses on the crosstalk cancellation problem for a stereo loudspeaker system. Least-squares methods are popular in designing a crosstalk cancellation system; however, the required large computation is always a challenge. To reduce the computational cost, this paper proposes a novel crosstalk cancellation system based on common-acoustical pole/zero (CAPZ) models, which outperforms conventional all-zero or pole/zero models in computational efficiency [23, 24]. The acoustic paths from loudspeakers to ears are approximated with CAPZ models, then the crosstalk cancellation filters are designed based on the CAPZ transfer functions. Compared with conventional least-squares methods, the proposed method can reduce the computation cost greatly. The paper is organized as follows. Conventional crosstalk cancellation methods are introduced in Section 2. Then the proposed crosstalk cancellation method based on the CAPZ model is described in detail in Section 3. The performance of the proposed method is evaluated in Section 4. Finally, conclusions are drawn in Section 5.

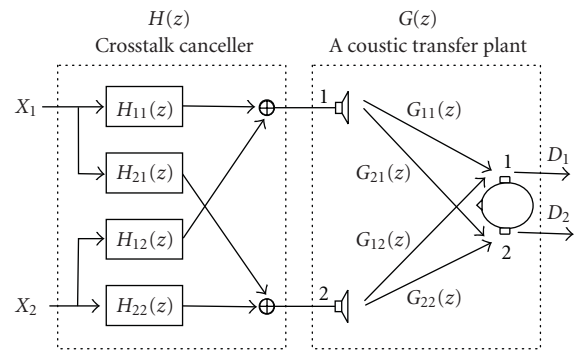


Figure 1: Block diagram of the direct crosstalk cancellation system for stereo loudspeakers.

2. Conventional Crosstalk Canceller

It is common to use two loudspeakers in a stereo system. A block diagram of the direct implementation of crosstalk cancellation is illustrated in Figure 1 for a stereo loudspeaker system. The input binaural signals from left and right channels are given in vector form $X(z) = [X_1(z), X_2(z)]^T$, and the signals received by two ears are denoted as $D(z) = [D_1(z), D_2(z)]^T$. (Here signals are expressed in the Z domain.) The objective of crosstalk cancellation is to perfectly reproduce the binaural signals at the listener's eardrums, that is, $D(z) = z^{-d}X(z)$, where z^{-d} is the delay term, via inverting the acoustic path $G(z)$ with the crosstalk cancellation filter $H(z)$. Generally, the loudspeaker response should also be inverted when designing the crosstalk canceller; however, this part can be implemented separately and thus is not considered in this paper for the convenience of analysis. $G(z)$ and $H(z)$ are, respectively, denoted in matrix forms as

$$G(z) = \begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix}, \quad H(z) = \begin{bmatrix} H_{11}(z) & H_{12}(z) \\ H_{21}(z) & H_{22}(z) \end{bmatrix}, \tag{1}$$

where $G_{ij}(z)$, $i, j = 1, 2$, is the acoustic transfer function from the jth loudspeaker to the ith ear, and $H_{ij}(z)$, $i, j = 1, 2$, is the crosstalk cancellation filter from X_j to the ith loudspeaker.

To ensure crosstalk cancellation, the global transfer function from binaural signals to ears should be

$$D(z) = G(z)H(z)X(z) = z^{-d}X(z), \tag{2}$$

thus

$$G(z)H(z) = z^{-d}I\ , \tag{3}$$

$$H(z) = z^{-d}G^{-1}(z), \tag{4}$$

where I is the identity matrix. The delay term z^{-d} is necessary to guarantee that $H(z)$ is physical realizable (causal). However, a perfect reproduction is impossible because $G(z)$ is generally nonminimum-phase, in which case a least-squares algorithm is employed to approximate the optimal inverse filter $G^{-1}(z)$. The time-domain least-squares algorithm is given below.

Suppose that $g_{ij} = [g_{ij,0}, \ldots, g_{ij,L_g-1}]^T$, the time-domain impulse response of $G_{ij}(z)$, is a vector of length L_g, and $h_{ij} = [h_{ij,0}, \ldots, h_{ij,L_h-1}]^T$, the time-domain impulse response of $H_{ij}(z)$, is a vector of length L_h. Rewriting (3) in a time-domain form, we get

$$\begin{bmatrix} \widetilde{G}_{11} & \widetilde{G}_{12} \\ \widetilde{G}_{21} & \widetilde{G}_{22} \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} u_d & O \\ O & u_d \end{bmatrix} \tag{5}$$

or in a suppressed form

$$GH = U, \tag{6}$$

where $\widetilde{G}_{ij}$, a component of G, is

$$\widetilde{G}_{ij} = \begin{bmatrix} g_{ij,0} & \cdots & g_{ij,L_g-1} & 0 & \cdots & 0 \\ 0 & g_{ij,0} & \cdots & g_{ij,L_g-1} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & g_{ij,0} & \cdots & g_{ij,L_g-1} \end{bmatrix}^T. \tag{7}$$

$\widetilde{G}_{ij}$ is a convolution matrix of size $L_1 \times L_h$ by cascading the vector g_{ij}, $L_1 = L_h + L_g - 1$,

$$u_d = [0, \ldots, 0, 1, 0, \ldots, 0]^T \tag{8}$$

is a vector of length L_1 whose dth component equals 1, and O is a vector of length L_1 containing only zeros.

The least-squares solution to (6) is

$$H_{\mathrm{LS}} = G^+ U, \tag{9}$$

where G^+ is the pseudoinverse of G, and G^+ is given by

$$G^+ = \left(G^T G + \beta I\right)^{-1} G^T, \tag{10}$$

where β is a regularization parameter to increase the robustness of the inversion [25].

The crosstalk cancellation filter is obtained by (9), with its filter length

$$L_{h1} = L_h. \tag{11}$$

The acoustic path matrix G is dependent on the head position. When the head moves, it is required to update G and calculate H in real time. The computation load becomes heavy when the size of G is large.

In [26], a single-filter structure for a stereo loudspeaker system is proposed to calculate the inverse of G, which needs less computation. It is given as follows.

From (4), we can get

$$\begin{aligned} H(z) &= z^{-d} G^{-1}(z) \\ &= \frac{z^{-d} \begin{bmatrix} G_{22}(z) & -G_{12}(z) \\ -G_{21}(z) & G_{11}(z) \end{bmatrix}}{G_{11}(z)G_{22}(z) - G_{12}(z)G_{21}(z)}. \end{aligned} \tag{12}$$

Let

$$Q(z) = G_{11}(z)G_{22}(z) - G_{12}(z)G_{21}(z), \tag{13}$$

$$T(z) = \frac{z^{-d}}{Q(z)}, \tag{14}$$

then the problem of inverting $G(z)$ is converted to

$$Q(z)T(z) = z^{-d} I. \tag{15}$$

Suppose that $q = [q_0, \ldots, q_{L_q-1}]^T$, the time-domain response of $Q(z)$, is a vector of length L_q, and $L_q = 2L_g - 1$; $t = [t_0, \ldots, t_{L_t-1}]^T$, the time-domain response of $T(z)$, is a vector of length L_t. Rewriting (15) in a time-domain form, we get

$$Qt = u_d, \tag{16}$$

where

$$Q = \begin{bmatrix} q_0 & \cdots & q_{L_q-1} & 0 & \cdots & 0 \\ 0 & q_0 & \cdots & q_{L_q-1} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & q_0 & \cdots & q_{L_q-1} \end{bmatrix}^T \tag{17}$$

is a convolution matrix of size $L_2 \times L_t$ by cascading of the vector q; $L_2 = L_t + L_q - 1$.

The least-squares solution to (16) is

$$t_{\mathrm{LS}} = Q^+ u_d, \tag{18}$$

where Q^+ is the pseudoinverse of Q, and Q^+ is given by

$$Q^+ = \left(Q^T Q + \beta I\right)^{-1} Q^T. \tag{19}$$

The crosstalk cancellation filter is obtained from (12) and (18), with its filter length

$$L_{h2} = L_t + L_g - 1. \tag{20}$$

Combining $G(z)$ and $H(z)$, we get the global transfer function

$$\begin{aligned} F(z) &= G(z) \cdot H(z) \\ &= T(z) \cdot \begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} \cdot \begin{bmatrix} G_{22}(z) & -G_{12}(z) \\ -G_{21}(z) & G_{11}(z) \end{bmatrix} \\ &= T(z) \\ &\quad \cdot \begin{bmatrix} G_{11}(z)G_{22}(z) & 0 \\ -G_{12}(z)G_{21}(z) & \\ 0 & G_{11}(z)G_{22}(z) \\ & -G_{12}(z)G_{21}(z) \end{bmatrix}. \end{aligned} \tag{21}$$

The off-diagonal items of (21) are always zeros regardless the value of $T(z)$. This implies that the crosstalk is almost fully suppressed. However, due to the filtering effect by the diagonal items in (21), distortion will be introduced when reproducing the target signals. This is the inherent disadvantage of the single-filter structure method.

3. Crosstalk Cancellation System Based on CAPZ Models

The acoustic transfer function is usually an all-zero model, whose coefficients are its impulse response. However, when the duration of the impulse response is long, it requires a large number of parameters to represent the transfer function [27]. This results in large computation in binaural synthesis and crosstalk cancellation. Pole/zero models may decrease the computational load, but their poles and zeros both change when the acoustic transfer function varies, leading to inconvenience for acoustic path inversion. To reduce the computational cost, this paper attempts to approximate the acoustic transfer function with common-acoustical pole/zero (CAPZ) models, then design a crosstalk cancellation system based on it.

3.1. CAPZ Modeling of Acoustic Transfer Functions. Haneda proposed the concept of common-acoustical pole/zero (CAPZ) models, and modeled room transfer functions and head-related transfer functions with good results [23, 24]. He believed that an HRTF contains a resonance system of ear canal whose resonance frequencies and Q factors are independent of source directions. Based on this, the HRTF can be efficiently modeled by using poles that are independent of source directions, with zeros that are dependent on source directions. The poles represent the resonance frequencies and Q factors. The model is called common-acoustical pole/zero model. CAPZ models share one set of poles and process their own zeros individually. This obviously reduces the amount of parameters with respect to conventional pole/zero models, and also cut down computation.

When an acoustic transfer function $H_i(z)$ is approximated with a CAPZ model, it is expressed as

$$\hat{H}_i(z) = \frac{B_i(\mathrm{z})}{A(z)} = \frac{\sum_{n=0}^{N_q} b_{n,i} z^{-n}}{1 + \sum_{n=1}^{N_p} a_n z^{-n}}, \tag{22}$$

where N_p and N_q are the numbers of the poles and zeros, $a = [1, a_1, \ldots, a_{N_p}]^T$ and $b_i = [b_{1,i}, \ldots, b_{N_q,i}]^T$ are the pole and zero coefficient vectors, respectively. The CAPZ parameters may be estimated with a least-squares method [23, 24] or a state-space method [28]. The least-squares method is simply given below.

Suppose a set of K transfer functions, the total modeling error is defined as

$$\begin{aligned} J &= \sum_{i=1}^{K} \sum_{n=0}^{N-1} |e_i(n)|^2 \\ &= \sum_{i=1}^{K} \sum_{n=0}^{N-1} \left| h_i(n) + \sum_{j=1}^{N_p} a_j h_i(n-j) - \sum_{j=0}^{N_q} b_{j,i} \delta(n) \right|^2, \end{aligned} \tag{23}$$

where N is the length of $e(n)$ and $h_i(n)$ is the impulse response of $H_i(z)$.

To find the pole coefficients vector a and the zero coefficients vector b_i, $i = 1, \ldots, K$, we minimize the error J and obtain that

$$\begin{aligned} \begin{bmatrix} I & H_{o,1} \\ 0 & H_1 \end{bmatrix} \begin{bmatrix} b_1 \\ -a \end{bmatrix} &= \begin{bmatrix} r_{o,1} \\ r_1 \end{bmatrix}, \\ &\vdots \\ \begin{bmatrix} I & H_{o,K} \\ 0 & H_K \end{bmatrix} \begin{bmatrix} b_K \\ -a \end{bmatrix} &= \begin{bmatrix} r_{o,K} \\ r_K \end{bmatrix}, \end{aligned} \tag{24}$$

where I is the identity matrix, vector $r_{o,i} = [h_i(0), \ldots, h_i(N_q)]^T$, $r_i = [h_i(N_q + 1), \ldots, h_i(N-1)]^T$, $i = 1, \ldots, K$; $H_{o,i}$ and H_i are both convolution matrices by cascading the impulse response $h_i(n)$, that is,

$$H_{o,i} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ h_i(0) & 0 & \cdots & 0 \\ h_i(1) & h_i(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_i(N_q - 1) & h_i(N_q - 2) & \vdots & h_i(N_q - N_p) \end{bmatrix}_{(N_q - 1) \times N_p}, \tag{25}$$

$$H_i = \begin{bmatrix} h_i(N_q) & \cdots & h_i(N_q - N_p + 1) \\ \vdots & \ddots & \vdots \\ h_i(N-2) & \cdots & h_i(N - 1 - N_p) \end{bmatrix}_{(N - 1 - N_q) \times N_p}. \tag{26}$$

From (24), a and b_i can be obtained by

$$\begin{aligned} a &= -\left(\tilde{H}^T \tilde{H}\right)^{-1} \tilde{H}^T R, \\ b_i &= H_{o,i} a + r_{o,i}, \qquad i = 1, \ldots, K, \end{aligned} \tag{27}$$

where vector $R = [r_1, \ldots, r_K]^T$ and matrix $\tilde{H} = [H_1, \ldots, H_K]^T$.

It is useful to specify the selection of the number of poles and zeros, N_p and N_q. The more poles and zeros used, the better approximation result may be obtained. On the other hand, more parameters require higher computation. Thus a trade-off should be considered. Generally, in the least-squares method, the number of parameters can be determined empirically [24]; or in the state-space method, it is determined based on the singular-value decomposition result [28].

3.2. Crosstalk Cancellation Based on the CAPZ Model. Supposing that acoustic transfer path G is known, the CAPZ

parameters are estimated. The CAPZ models from the loudspeakers to the ears are

$$\begin{aligned} G_{11}(z) &= \frac{B_{11}(z)}{A(z)}z^{-d_{11}}, \\ G_{12}(z) &= \frac{B_{12}(z)}{A(z)}z^{-d_{12}}, \\ G_{21}(z) &= \frac{B_{21}(z)}{A(z)}z^{-d_{21}}, \\ G_{22}(z) &= \frac{B_{22}(z)}{A(z)}z^{-d_{22}}, \end{aligned} \tag{28}$$

where d_{11}, d_{12}, d_{21}, and d_{22} are the transmission delays from the loudspeakers to the ears.

Substituting (28) into (4), we get

$$\begin{aligned} &H(z) \\ &= z^{-d}G^{-1}(z) \\ &= \frac{z^{-d}\begin{bmatrix} G_{22}(z) & -G_{12}(z) \\ -G_{21}(z) & G_{11}(z) \end{bmatrix}}{G_{11}(z)G_{22}(z) - G_{12}(z)G_{21}(z)} \\ &= z^{-d}/\left(\left(\frac{B_{11}(z)B_{22}(z)}{A^2(z)}\right)z^{-(d_{11}+d_{22})}\right. \\ &\qquad \left. -\left(\frac{B_{12}(\mathrm{z})B_{21}(\mathrm{z})}{A^2(z)}\right)z^{-(d_{12}+d_{21})}\right) \\ &\quad \times \begin{bmatrix} \left(\frac{B_{22}(z)}{A(z)}\right)z^{-d_{22}} & \left(-\frac{B_{12}(z)}{A(z)}\right)z^{-d_{12}} \\ \left(-\frac{B_{21}(z)}{A(z)}\right)z^{-d_{21}} & \left(\frac{B_{11}(z)}{A(z)}\right)z^{-d_{11}} \end{bmatrix} \\ &= \frac{z^{-d}}{B_{11}(z)B_{22}(z)z^{-(d_{11}+d_{22})} - B_{12}(z)B_{21}(z)z^{-(d_{12}+d_{21})}} \\ &\quad \times \begin{bmatrix} B_{22}(z)A(z)z^{-d_{22}} & -B_{12}(z)A(z)z^{-d_{12}} \\ -B_{21}(z)A(z)z^{-d_{21}} & B_{11}(z)A(z)z^{-d_{11}} \end{bmatrix}. \end{aligned} \tag{29}$$

Without loss of generality, assume $d_{11} + d_{22} < d_{12} + d_{21}$, and let $\Delta = (d_{11} + d_{22}) - (d_{12} + d_{21})$. Substituting Δ into (29), we get

$$\begin{aligned} H(z) &= \frac{z^{-(d-d_{11}-d_{22})}}{B_{11}(z)B_{22}(z) - B_{12}(z)B_{21}(z)z^{-\Delta}} \\ &\quad \times \begin{bmatrix} B_{22}(z)A(z)z^{-d_{22}} & -B_{12}(z)A(z)z^{-d_{12}} \\ -B_{21}(z)A(z)z^{-d_{21}} & B_{22}(z)A(z)z^{-d_{11}} \end{bmatrix} \\ &= \frac{z^{-\delta}}{B(z)}\begin{bmatrix} B_{22}(z)A(z)z^{-d_{22}} & -B_{12}(z)A(z)z^{-d_{12}} \\ -B_{21}(z)A(z)z^{-d_{21}} & B_{22}(z)A(z)z^{-d_{11}} \end{bmatrix} \\ &= C(z)\begin{bmatrix} B_{22}(z)A(z)z^{-d_{22}} & -B_{12}(z)A(z)z^{-d_{12}} \\ -B_{21}(z)A(z)z^{-d_{21}} & B_{11}(z)A(z)z^{-d_{11}} \end{bmatrix}, \end{aligned} \tag{30}$$

where $B(z) = B_{11}(z)B_{22}(z) - B_{12}(z)B_{21}(z)z^{-\Delta}$, $C(z) = z^{-\delta}/B(z)$, and $\delta = d - (d_{11} + d_{22})$ is the delay.

Thus the problem of inverting $G(z)$ is converted to

$$B(z)C(z) = z^{-\delta}I. \tag{31}$$

Suppose that $b = [b_0, \ldots, b_{L_b-1}]^T$, the time-domain impulse response of $B(z)$, is a vector of length L_b, and $L_b = 2(N_q + 1) + \Delta - 1$; $c = [c_0, \ldots, c_{L_c-1}]^T$, the time-domain impulse response of $C(z)$, is a vector of length L_c. Rewriting (31) in a time-domain form, we get

$$Bc = u_\delta, \tag{32}$$

where B is a convolution matrix of size $L_3 \times L_c$ by cascading the vector b, and $L_3 = L_b + L_c - 1$,

$$\begin{aligned} B &= \begin{bmatrix} b_0 & \ldots & b_{L_b-1} & 0 & \ldots & 0 \\ 0 & b_0 & \ldots & b_{L_b-1} & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & b_0 & \ldots & b_{L_b-1} \end{bmatrix}^T, \\ u_\delta &= [0, \ldots, 0, 1, 0, \ldots, 0]^T \end{aligned} \tag{33}$$

is a vector of length L_3 whose δth component equas 1.

Since $B(z)$ is generally nonminimum-phase, the least-squares solution to (32) is

$$c_{\mathrm{LS}} = B^+ u_\delta, \tag{34}$$

where B^+ is the pseudoinverse of B, and B^+ is given by

$$B^+ = \left(B^T B + \beta I\right)^{-1} B^T, \tag{35}$$

where β is the regularization parameter.

Finally, the crosstalk canceller is obtained by (30) and (34), with its filter length

$$\begin{aligned} L_{h3} &= L_c + \left(N_q + 1\right) + \left(N_p + 1\right) + \max(d_{11}, d_{12}, d_{21}, d_{22}) - 1 \\ &= L_c + N_q + N_p + d_{\max} + 1, \end{aligned} \tag{36}$$

where $d_{\max} = \max(d_{11}, d_{12}, d_{21}, d_{22})$.

3.3. Computational Complexity Analysis. Now we discuss the computational complexity of the three methods (the least-squares method, the single-filter structure method, and the CAPZ method) from two aspects: crosstalk cancellation filter estimation and implementation. For the convenience of comparison, Table 1 lists some parameters for three methods, respectively, where the column "Inverse filter" denotes the filter resulted from matrix inversion (referring to (9), (18), and (34)), the column "Matrix size" denotes the size of the matrix being inverted. It should be noted that the term "inverse filter" is different from the term "crosstalk cancellation filter."

Table 1: Parameters for the three methods: the least-squares method, the single-filter structure method, and the CAPZ method.

Method	Inverse filter	Matrix size	Crosstalk cancellation filter length
Least-squares	h	$Size(G) = 2L_1 \times 2L_h$	$L_{h1} = L_h$
Single-filter structure	t	$Size(Q) = L_2 \times L_t$	$L_{h2} = L_t + L_g - 1$
CAPZ	c	$Size(B) = L_3 \times L_c$	$L_{h3} = L_c + N_p + N_p + d_{\max} + 1$

Table 2: Computational complexity of crosstalk cancellation filter estimation for the three methods: the least-squares method, the single-filter structure method, and the CAPZ method.

Method	Computation cost (in multiplications)
Least-squares	$8(O(L_{\text{inv}}^3) + 2L_{\text{inv}}^2 L_1)$
Single-filter structure	$O(L_{\text{inv}}^3) + 2L_{\text{inv}}^2 L_2$
CAPZ	$O(L_{\text{inv}}^3) + 2L_{\text{inv}}^2 L_3$

3.3.1. Computational Complexity of Crosstalk Cancellation Filter Estimation. From (9), (12), and (30), it is found that estimating the inverse filters h, t, and c consumes the major computation of crosstalk cancellation filter estimation. Thus only the computation of calculating the inverse filters is considered. Generally, the computational complexity of inverting a matrix of size $N \times N$ is $O(N^3)$, without taking advantage of matrix symmetry. The computation of estimating the inverse filters h, t, and c is closely related to the size of the matrix G, Q, and B, respectively. Supposing that the inverse filter lengths in the three methods are equal, that is, $L_h = L_t = L_b = L_{\text{inv}}$, we summarize the computational complexity in Table 2 for the three methods (referring to (9), (18), and (34)). The computational complexity is calculated in terms of multiplication. For example, when the size of G is $2L_1 \times 2L_h$, the number of calculations involved in matrix multiplication is $16L_h^2 L_1$, and matrix inversion is $O((2L_h)^3)$ (referring to (9), (10), and Table 1). Thus, the computation cost of the least-squares method is $8(O(L_h^3) + 2L_h^2 L_1)$, as listed in Table 2. The computation cost of the other two methods can be obtained in a similar way.

For the convenience of comparison, we rewrite the parameters L_1, L_2, and L_3 from Table 1 in an approximated form as

$$\begin{aligned} L_1 &= L_h + L_g - 1 \approx L_{\text{inv}} + L_g, \\ L_2 &= L_t + L_q - 1 = L_t + 2L_g - 2 \approx L_{\text{inv}} + 2L_g, \\ L_3 &= L_c + L_b - 1 = L_c + 2N_q + \Delta \approx L_{\text{inv}} + 2N_q. \end{aligned} \tag{37}$$

Generally, $L_g \gg N_q$ holds for a CAPZ model. Thus we have

$$L_2 > L_1 > L_3. \tag{38}$$

From Table 2, the computational complexity of the least-squares method is much higher than the other two methods (almost 8 times), while the computation of the single-filter structure method is a little higher than the proposed CAPZ method.

3.3.2. Computational Complexity of Crosstalk Cancellation Filter Implementation. The computational complexity of crosstalk cancellation implementation is proportional to the crosstalk cancellation filter length, as listed in Table 1. Since $L_g > N_p + N_q + d_{\max}$ holds for the CAPZ model, we have

$$L_{h1} < L_{h3} < L_{h2}, \tag{39}$$

with the assumption of $L_h = L_t = L_b$.

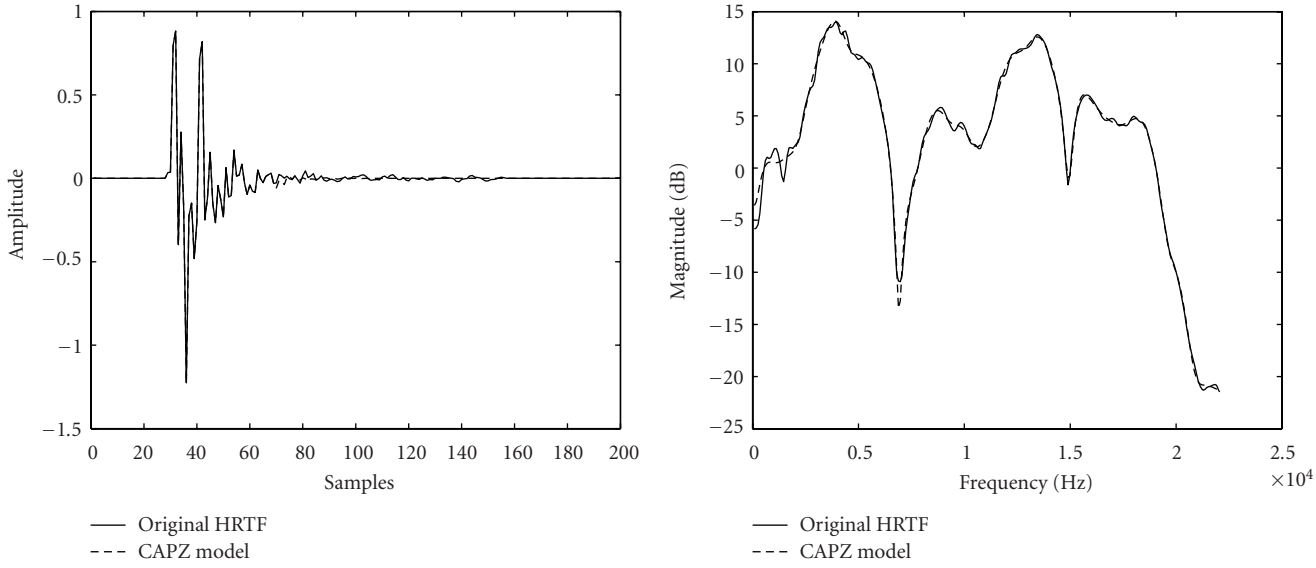
The least-squares method has the lowest computational complexity in crosstalk cancellation filter implementation, while the single-filter structure method has the highest one.

In summary, although the least-squares method has the lowest computational cost in filter implementation, its complexity in filter estimation is much higher than the other two. On the other hand, the CAPZ method has the lowest complexity in filter estimation, and ranks second in terms of the complexity of filter implementation. In a global view of both measures, the CAPZ method is the most effective among the three ones. Later, the performance comparison of the three methods will be carried out in Section 4.3 under the same assumption with $L_h = L_t = L_b = L_{\text{inv}}$.

4. Performance Evaluation

The acoustic transfer function can be estimated based on the positions of loudspeakers and ears. Head-related transfer functions (HRTF) provide a measure of the transfer path of a sound from some point in space to the ear canal. This paper assumes that the acoustic transfer function can be represented by HRTF in anechoic conditions. The HRTFs used in our experiments are from the extensive set of HRTFs measured at the CIPIC Interface Laboratory, University of California [29]. The database is composed of HRTFs for 45 subjects, and each subject contains 1250 HRTFs measured at 25 different azimuths and 50 different elevations. The HRTF is 200 taps long with a sampling rate of 44.1 kHz. In the experiment, the HRTFs are modeled as CAPZ models first, then the performance of the proposed crosstalk cancellation method is evaluated in two cases for loudspeakers placement: symmetric and asymmetric cases.

4.1. Experiments on CAPZ Modeling. For subject "003", the HRTFs from all 1250 positions are approximated with CAPZ models. Before modeling, the initial delay of each HRIR is recorded and removed. The common pole number is set empirically as $N_p = 20$, and the zero number $N_q = 40$. The original and modeled impulse responses and magnitude responses of the right ear HRTF at elevation 0°, azimuth 30° are shown in Figures 2(a) and 2(b), respectively. It can be seen from these figures that only small distortions can be noticed between the original and modeled HRTFs. Similar results may be observed at other HRTF positions.



(a) Impulse responses of the original and modeled HRTFs

(b) Magnitude responses of the original and modeled HRTFs

Figure 2: Comparison of the original and modeled right ear HRTF at elevation 0°, azimuth 30°.

4.2. Performance Metrics. Two performance measures are used: the signal-to-crosstalk ratio (SCR) and the signal-to-distortion ratio (SDR) [8]. Regarding to (6), the ideal crosstalk cancellation result should be

$$\mathrm{GH} = U = \begin{bmatrix} u_1 & O \\ O & u_2 \end{bmatrix}. \tag{40}$$

Since G is generally nonminimum-phase, the actual crosstalk cancellation result is

$$\mathrm{GH} = F = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}. \tag{41}$$

The signal-to-crosstalk ratio at two ears would be

$$\mathrm{SCR}_1 = \frac{f_{11}^T f_{11}}{f_{12}^T f_{12}}, \quad \mathrm{SCR}_2 = \frac{f_{22}^T f_{22}}{f_{21}^T f_{21}}, \tag{42}$$

and the average signal-to-crosstalk ratio is given by $\mathrm{SCR} = (\mathrm{SCR}_1 + \mathrm{SCR}_2)/2$.

And the signal-to-distortion ratio at two ears is determined by

$$\begin{aligned} \mathrm{SDR}_1 &= \frac{1}{(f_{11} - u_1)^T (f_{11} - u_1)}, \\ \mathrm{SDR}_2 &= \frac{1}{(f_{22} - u_2)^T (f_{22} - u_2)}, \end{aligned} \tag{43}$$

and the average signal-to-distortion ratio is $\mathrm{SDR} = (\mathrm{SDR}_1 + \mathrm{SDR}_2)/2$.

According to the definitions above, the signal-to-crosstalk ratio measures the crosstalk suppression performance, and signal-to-distortion ratio measures the signal reproduction performance.

4.3. Performance Evaluation in Symmetric Cases. In this experiment, the loudspeakers are placed in symmetric positions. Three crosstalk cancellation methods are compared: the least-squares method, the single-filter structure method, and the proposed method based on CAPZ models. To be consistent with the assumption in computational complexity analysis in Section 3.3, the inverse filter lengths in the three methods are set equal, that is, $L_h = L_t = L_c$. A total of 63 crosstalk cancellation systems are designed at 7 different elevations uniformly spaced between 0° and 67.5° and 9 different azimuths uniformly spaced between 5° and 45°. For each crosstalk cancellation system, various inverse filter lengths ranging from 50 to 400 samples with an interval of 50 are tested. Generally, the crosstalk cancellation performance is not quite sensitive to the delay value; however, an optimal delay value is selected for each method separately so that they can be compared in a fair condition. Since the relationship between the crosstalk cancellation and the delay z^{-d} shows no evident regularity, we choose the delay value experimentally. For each experiment case, the optimal delay is selected experimentally from values ranging from 50 to 400 samples with an interval of 50, ensuring that the crosstalk cancellation algorithm performs best with this optimal delay. Table 3 lists the optimal delay for the three methods at various inverse filter lengths. The regularization parameter is set empirically as $\beta = 0.005$ throughout the experiment. The mean value of the performance metrics over all 63 crosstalk cancellation systems is calculated.

Figure 3 shows the mean signal-to-distortion ratio (SDR), respectively, for the three methods with various inverse filter lengths. The horizontal axis is the inverse filter length ranging from 50 to 400 samples. The vertical axis is the mean signal-to-distortion ratio. The SDR of the least-squares method is always 2-3 dB higher than the CAPZ method, and 3-5 dB higher than the single-filter structure method.

Table 3: Optimal delay d at various inverse filter lengths (in samples) for the three methods: the least-squares method (LS), the single-filter structure method (SF), and the CAPZ method.

Filter length	LS	SF	CAPZ
50	50	100	100
100	100	150	150
150	100	150	150
200	150	200	200
250	150	200	200
300	200	250	250
350	200	250	250
400	250	300	300

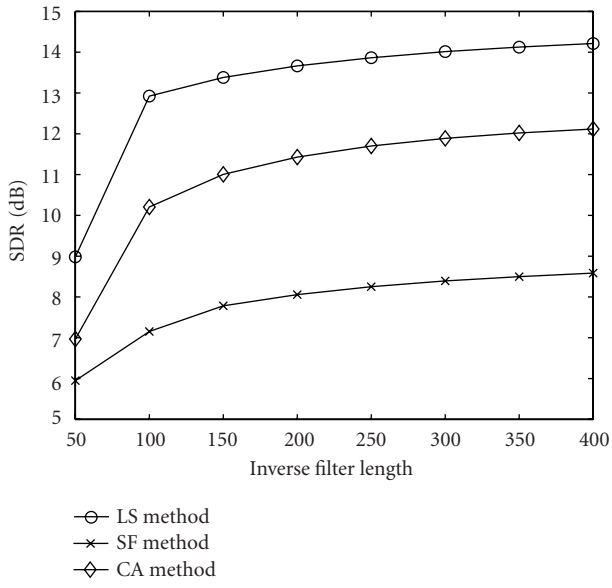


Figure 3: Mean signal-to-distortion ratio (SDR) at different inverse filter lengths for the three methods: the least-squares method (LS), the single-filter structure method (SF), and the CAPZ method.

Figure 4 shows the mean signal-to-crosstalk ratio (SCR), respectively, for the three methods with various inverse filter lengths. The horizontal axis is the inverse filter length ranging from 50 to 400 samples. The vertical axis is the mean signal-to-crosstalk ratio. Since the SCR of the SF method can be as high as 300 dB for all simulation cases, which is much higher than the levels of the other two methods (20–30 dB), its curve is left out of the picture. The SCR of the CAPZ is higher than the least-squares method. It can be seen from Figures 3 and 4 that the single-filter structure method yields the best SCR performance, while the least-squares method yields best SDR performance. On the other hand, for both SDR and SCR measures, the proposed CAPZ method yields performance that is superior to one of the reference methods, but inferior to the other reference. In a view of crosstalk cancellation, the performance of the CAPZ method is in the middle of the three methods. It can yield comparable crosstalk cancellation as the other two methods do.

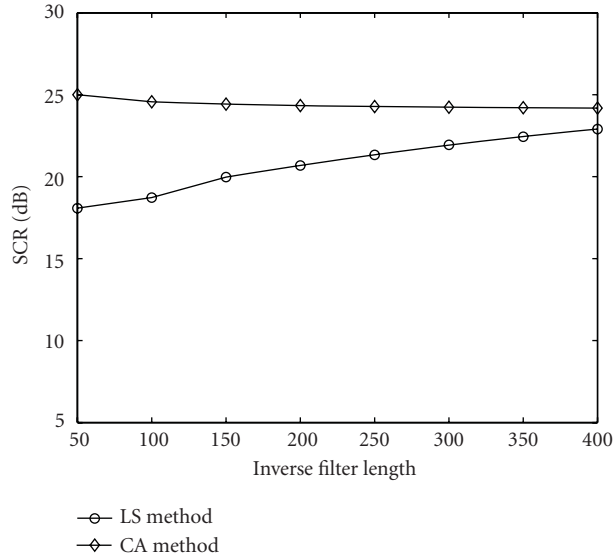


Figure 4: Mean signal-to-crosstalk ratio (SCR) at different inverse filter lengths for the three methods: the least-squares method (LS), the single-filter structure method (SF), and the CAPZ method. (Note that the curve of the SF method is not depicted in the picture, because its SCR values can be as high as 300 dB for all simulation cases.)

As discussed at the end of Section 2, with the off-diagonal items of the global transfer function (21) being zeros, the single-filter structure method can obtain nearly perfect crosstalk suppression. That is why the signal-to-crosstalk ratio (SCR) can be as high as 300 dB, which is implied in Figure 4. In practice, inevitable errors in the measurement process (nonideal HRTFs) result in degraded performance. To conduct a more realistic evaluation, we add random white noises with a signal-to-noise ratio of 30 dB to the HRTF measurement, and repeat the previous experiment. Although this is not a real non-ideal HRTF, the white noise may partly simulate errors and disturbances encountered during the measurement. This process is repeated five times, and then an average result is calculated. The mean signal-to-distortion ratio and signal-to-crosstalk ratio of the three methods are shown in Figures 5 and 6, respectively. The result is similar to the noise-free case: the performance of the three methods all decreases a little; especially, the SCR of the single-filter structure method reduce to about 26 dB.

From Figures 3–6, similar variation trends of the signal-to-distortion ratio (SDR) and signal-to-crosstalk ratio (SCR) may be observed for both noisy and noise-free cases. For all the three methods, the SDR performance increases with the inverse filter length L_{inv}, and the increase is small for $L_{\text{inv}} > 150$. The slow variation of SDR for large L_{inv} may be related to the least-squares matrix inversion process. When L_{inv} increases, the size of the matrices G, Q and B increases, the matrix inversion becomes difficult and more errors will be introduced. The error may cancel part of the benefit brought by a longer inverse filter. Thus the SDR increases slowly for large inverse filter length. With regard to the SCR performance, the least-squares method yields increasing SCR

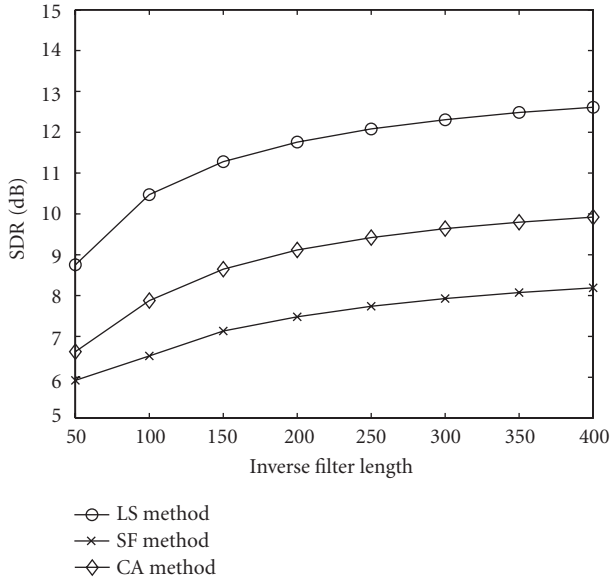


Figure 5: Mean signal-to-distortion ratio (SDR) at different inverse filter lengths for the three methods: the least-squares method (LS), the single-filter structure method (SF), and the CAPZ method (white noise added to HRTF).

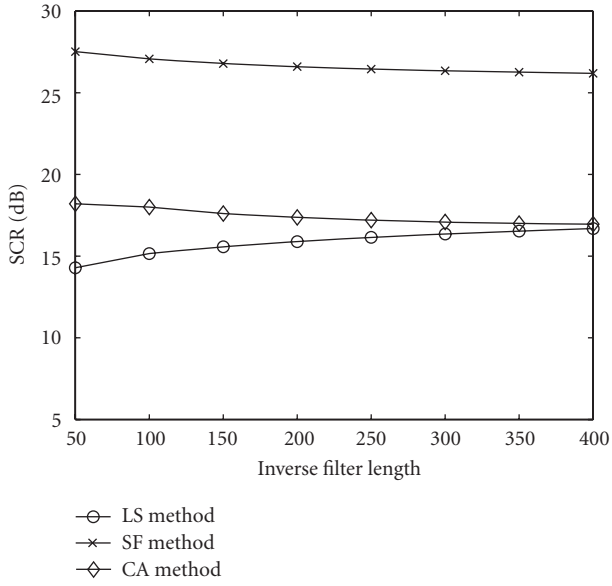


Figure 6: Mean signal-to-crosstalk ratio (SCR) at different inverse filter lengths for the three methods: the least-squares method (LS), the single-filter structure method (SF), and the CAPZ method (white noise added to HRTF).

with the increasing inverse filter length, while the single-filter structure method and the CAPZ method yield almost constant SCR with the increasing inverse filter length. Since the off-diagonal items of (21) are always zeros regardless of the value of $T(z)$, the SCR of the single-filter structure method is little affected by the inverse filter length. Likewise, the CAPZ method shows similar trend as the single-filter structure method does. In Figure 6, a slow decrease is also noticed for the curves of the CAPZ method and the single-filter structure method, which may be caused by the noise added to the acoustic transfer functions.

Table 4: Mean crosstalk cancellation performance in the symmetric case for the three methods when the inverse filter length equals 150.

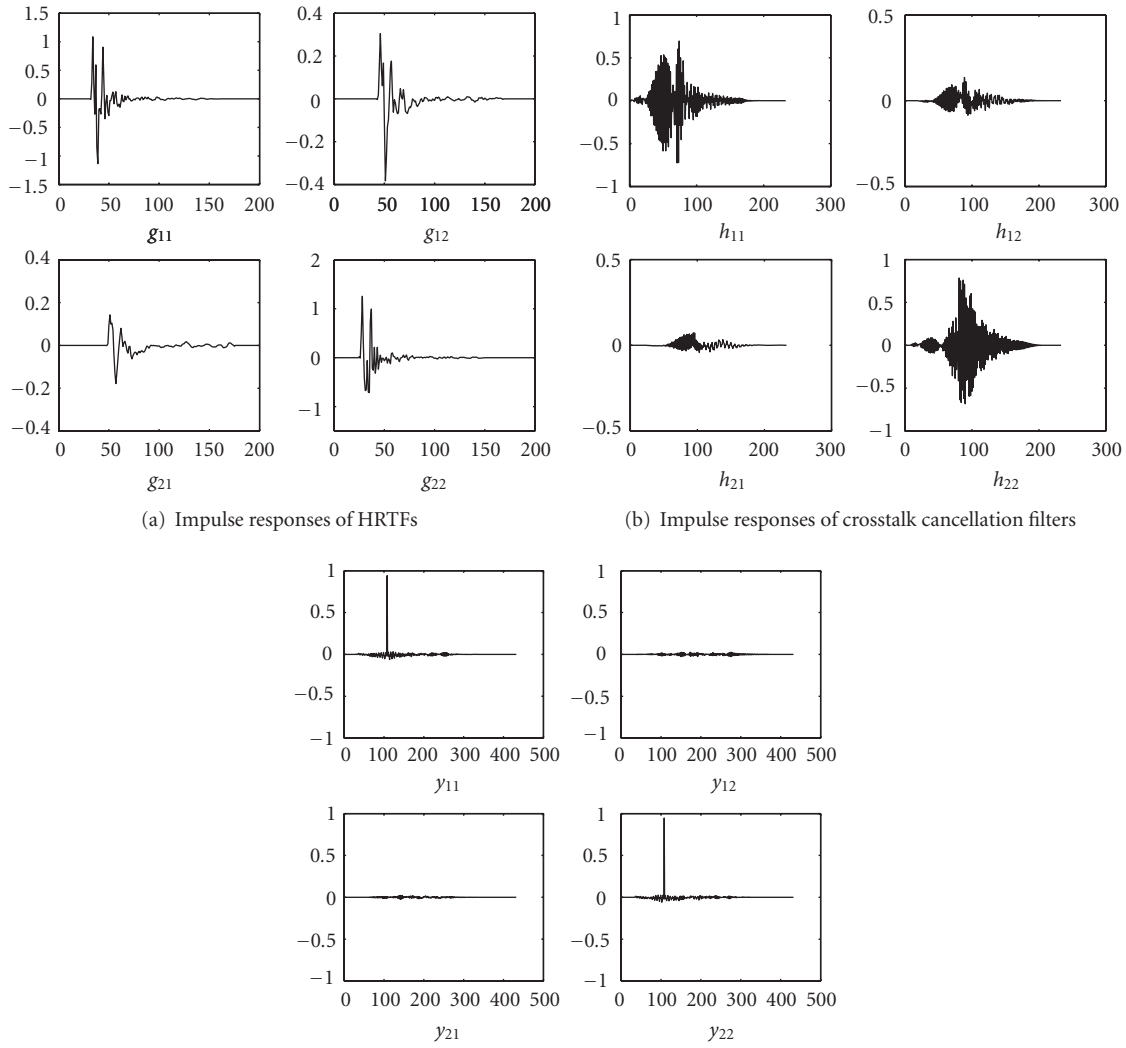
Method	SDR(dB)	SCR(dB)	Crosstalk cancellation filter length
Least-squares	11.2	15.6	150
Single-filter structure	7.1	26.8	349
CAPZ	8.6	17.6	233

Table 5: Crosstalk cancellation performance in the asymmetric case for the three methods when the inverse filter length equals 150.

Method	SDR(dB)	SCR(dB)
Least-squares	14.7	18.9
Single-filter structure	10.2	27.7
CAPZ	12.0	19.1

In summary, the proposed CAPZ method yields similar crosstalk cancellation performance as the other two methods do, meanwhile it is more computationally efficient. In a global view of both crosstalk cancellation and computational complexity, the proposed method is superior to the other two methods. Taking both performance and computation into consideration, we set the inverse filter length at 150. When white noises with a signal-to-noise ratio of 30 dB is added to HRTF, the performance of the three methods are listed in Table 4. The result in Table 4 also verifies the conclusion above.

4.4. Performance Evaluation in Asymmetric Cases. In this experiment, the stereo loudspeakers are placed in asymmetric positions, with the left and right loudspeakers at 30° and 60°, respectively, equidistant from the listener. Although this is not a common audio system, the crosstalk canceller can reproduce the desired sound field around the listener. The inverse filter length is set at 150, the regularization parameter is set at $\beta = 0.005$, the filter delay d is chosen from Table 3, white noise with a signal-to-noise ratio of 30 dB is added to the HRTF measurement. The performance of the three methods is shown in Table 5. Comparing Table 4 with Table 5, it can be seen that the performance of the three methods in the asymmetric cases is similar to that in the symmetric case. To give the readers a better understanding of the principle of crosstalk cancellation, Figure 7 depicts the impulse responses of the crosstalk cancellation system by the CAPZ method. The impulse responses of the HRTFs of 200 taps are shown in Figure 7(a), the four crosstalk cancellation filters designed by the CAPZ method are shown in Figure 7(b), and the result impulse responses after crosstalk cancellation are shown in Figure 7(c). Clearly, a good crosstalk cancellation can be obtained.



(a) Impulse responses of HRTFs

(b) Impulse responses of crosstalk cancellation filters

(c) Resulted impulse responses after crosstalk cancellation

Figure 7: Impulse responses of crosstalk cancellation in the asymmetric case.

5. Conclusion

This paper investigates crosstalk cancellation for authentic binaural reproduction of stereo sounds over two loudspeakers. Since the crosstalk cancellation filter has to be updated according to the head position in real time, the computational efficiency of the crosstalk cancellation algorithm is crucial for practical applications. To reduce the computational cost, this paper presents a novel crosstalk cancellation system based on common-acoustical pole/zero (CAPZ) models. The acoustic transfer paths from loudspeakers to ears are approximated with CAPZ models, then the crosstalk cancellation filter is designed based on the CAPZ model. Since the CAPZ model has advantages in storage and computation, the proposed method is more efficient than conventional ones. Simulation results demonstrate that the proposed method can reduce the computational complexity greatly with comparable crosstalk cancellation performance with respect to conventional methods.

The experiment in this paper is conducted in anechoic conditions. However, with promising results in anechoic environments, the proposed method can be extended to realistic situations. For example, in reverberation conditions, the acoustic transfer functions may also be approximated by the CAPZ model, and then crosstalk cancellation may be conducted in a similar way. However, due to large computational complexity and time-varying environments, this situation has not been specially addressed. Our further research will focus on this practical problem.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (60772161, 60372082) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (200801410015). This work is also supported by NRC-MOE Research and Postdoctoral Fellowship

Program from Ministry of Education of China and National Research Council of Canada.The authors gratefully acknowledge stimulating discussions with Dr. Heping Ding and Dr. Michael R. Stinson from Institute for Microstructural Sciences, National Research Council Canada.

References

[1] D. R. Begault, *3D Sound for Virtual Reality and Multimedia*, Academic Press, London, UK, 1st edition, 1994.

[2] A. W. Bronkhorst, "Localization of real and virtual sound sources," *Journal of the Acoustical Society of America*, vol. 98, no. 5, pp. 2542–2553, 1995.

[3] W. G. Gardner and K. D. Martin, "HRTF measurements of a KEMAR," *Journal of the Acoustical Society of America*, vol. 97, no. 6, pp. 3907–3908, 1995.

[4] M. Otani and S. Ise, "Fast calculation system specialized for head-related transfer function based on boundary element method," *Journal of the Acoustical Society of America*, vol. 119, no. 5, pp. 2589–2598, 2006.

[5] B. S. Atal and M. R. Schroeder, "Apparent sound source translator," US Patent no. 3,236,949, 1966.

[6] B. B. Bauer, "Stereophonic earphones and binaural loudspeakers," *Journal of the AudioEngineering Society*, vol. 9, no. 2, pp. 148–151, 1961.

[7] O. Kirkeby, P. A. Nelson, H. Hamada, and F. Orduna-Bustamante, "Fast deconvolution of multichannel systems using regularization," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 189–194, 1998.

[8] Y. Huang, J. Benesty, and J. Chen, "On crosstalk cancellation and equalization with multiple loudspeakers for 3-D sound reproduction," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 649–652, 2007.

[9] J. Garas, *Adaptive 3D Sound Systems*, Kluwer Academic Publishers, Norwell, Mass, USA, 2000.

[10] A. Mouchtaris, P. Reveliotis, and C. Kyriakakis, "Inverse filter design for immersive audio rendering over loudspeakers," *IEEE Transactions on Multimedia*, vol. 2, no. 2, pp. 77–87, 2000.

[11] P. A. Nelson, H. Hamada, and S. J. Elliott, "Adaptive inverse filters for stereophonic sound reproduction," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1621–1632, 1992.

[12] A. Gonzalez and J. J. Lopez, "Time domain recursive deconvolution in sound reproduction," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 833–836, June 2000.

[13] S. M. Kuo and G. H. Canfield, "Dual-channel audio equalization and cross-talk cancellation for 3-D sound reproduction," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 4, pp. 1189–1196, 1997.

[14] C. Kyriakakis, "Fundamental and Technological Limitations of Immersive Audio Systems," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 941–951, 1998.

[15] M. R. Bai and C.-C. Lee, "Objective and subjective analysis of effects of listening angle on crosstalk cancellation in spatial sound reproduction," *Journal of the Acoustical Society of America*, vol. 120, no. 4, pp. 1976–1989, 2006.

[16] T. Lentz, "Dynamic crosstalk cancellation for binaural synthesis in virtual reality environments," *Journal of the Audio Engineering Society*, vol. 54, no. 4, pp. 283–294, 2006.

[17] D. B. Ward and G. W. Elko, "Effect of loudspeaker position on the robustness of acoustic crosstalk cancellation," *IEEE Signal Processing Letters*, vol. 6, no. 5, pp. 106–108, 1999.

[18] T. Takeuchi and P. A. Nelson, "Optimal source distribution for binaural synthesis over loudspeakers," *Journal of the Acoustical Society of America*, vol. 112, no. 6, pp. 2786–2797, 2002.

[19] M. R. Bai, C.-W. Tung, and C.-C. Lee, "Optimal design of loudspeaker arrays for robust cross-talk cancellation using the Taguchi method and the genetic algorithm," *Journal of the Acoustical Society of America*, vol. 117, no. 5, pp. 2802–2813, 2005.

[20] J. Yang, W.-S. Gan, and S.-E. Tan, "Improved sound separation using three loudspeakers," *Acoustic Research Letters Online*, vol. 4, pp. 47–52, 2003.

[21] Y. Kim, O. Deille, and P. A. Nelson, "Crosstalk cancellation in virtual acoustic imaging systems for multiple listeners," *Journal of Sound and Vibration*, vol. 297, no. 1-2, pp. 251–266, 2006.

[22] M. Kallinger and A. Mertins, "A spatially robust least squares crosstalk canceller," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, pp. 177–180, April 2007.

[23] Y. Haneda, S. Makino, and Y. Kaneda, "Common acoustical pole and zero modeling of room transfer functions," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 320–328, 1994.

[24] Y. Haneda, S. Makino, Y. Kaneda, and N. Kitawaki, "Common-acoustical-pole and zero modeling of head-related transfer functions," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 188–195, 1999.

[25] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.

[26] S.-M. Kim and S. Wang, "A Wiener filter approach to the binaural reproduction of stereo sound," *Journal of the Acoustical Society of America*, vol. 114, no. 6, pp. 3179–3188, 2003.

[27] L. Wang, F. Yin, and Z. Chen, "HRTF compression via principal components analysis and vector quantization," *IEICE Electronics Express*, vol. 5, no. 9, pp. 321–325, 2008.

[28] D. W. Grantham, J. A. Willhite, K. D. Frampton, and D. H. Ashmead, "Reduced order modeling of head related impulse responses for virtual acoustic displays," *Journal of the Acoustical Society of America*, vol. 117, no. 5, pp. 3116–3125, 2005.

[29] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF database," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 99–102, October 2001.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 415840, 13 pages
doi:10.1155/2010/415840

Research Article

A Sparsity-Based Approach to 3D Binaural Sound Synthesis Using Time-Frequency Array Processing

Maximo Cobos,[1] Jose J. Lopez (EURASIP Member),[1] and Sascha Spors (EURASIP Member)[2]

[1] *Institute of Telecommunications and Multimedia Applications, Universidad Politécnica de Valencia, 46022 Valencia, Spain*
[2] *Deutsche Telekom Laboratories, Technische Universität Berlin, 10578 Berlin, Germany*

Correspondence should be addressed to Maximo Cobos, mcobos@iteam.upv.es

Received 2 March 2010; Revised 21 June 2010; Accepted 7 September 2010

Academic Editor: Augusto Sarti

Localization of sounds in physical space plays a very important role in multiple audio-related disciplines, such as music, telecommunications, and audiovisual productions. Binaural recording is the most commonly used method to provide an immersive sound experience by means of headphone reproduction. However, it requires a very specific recording setup using high-fidelity microphones mounted in a dummy head. In this paper, we present a novel processing framework for binaural sound recording and reproduction that avoids the use of dummy heads, which is specially suitable for immersive teleconferencing applications. The method is based on a time-frequency analysis of the spatial properties of the sound picked up by a simple tetrahedral microphone array, assuming source sparseness. The experiments carried out using simulations and a real-time prototype confirm the validity of the proposed approach.

1. Introduction

Human hearing plays a major role in the way our environment is perceived. Generally, sound is perceived in all three dimensions, width, height, and depth, which are all necessary to achieve a natural perception of sound [1]. These attributes are usually employed to describe the spatial characteristics of sound taking into account its diffuseness properties. The human auditory system is very sophisticated and, thus, capable to analyze and extract most spatial information pertaining to a sound source using two ears. In fact, when a sound scene is recorded by a single microphone, we are still able to recognize the original sound events. However, much of the information corresponding to the spatial properties of these events is lost. As a result, spatial sound recording and reproduction techniques are always based on a multichannel approach.

Reproduction using two-channels or *stereo* is the most common way that most people know to convey some spatial content into sound recording and reproduction, and this can be considered as the simplest approximation to spatial sound. On the other hand, *surround sound* systems have evolved and entered homes in order to give a better sensation than stereo by using more reproduction channels and have been widely utilized in theaters since the middle 70s. Both stereo and surround systems have an optimal listening position, known as *sweet spot* [2]. This optimum listening area is almost limited to the central point in the loudspeaker setup. Outside the central zone, the perceived virtual source locations differ significantly from their intended spatial position.

Another much more realistic strategy is to reproduce directly in the ears of the listener, via headphones, the signal that he/she would perceive in the acoustic environment that is intended to be simulated. This strategy is widely known as *binaural reproduction*. The signals to be reproduced with headphones can be recorded with an acoustic *dummy head* or they can be artificially synthesized by using a measured *Head-Related Transfer-Function* (HRTF) [3]. In an anechoic environment, as sound propagates from the source to the listener, its own head, pinna, and torso introduce changes to the sound before it reaches the ear drums. These effects of the listener's body are registered by the HRTF, which is the transfer function between the sound pressure that is present at the center of the listener's head when the listener is absent

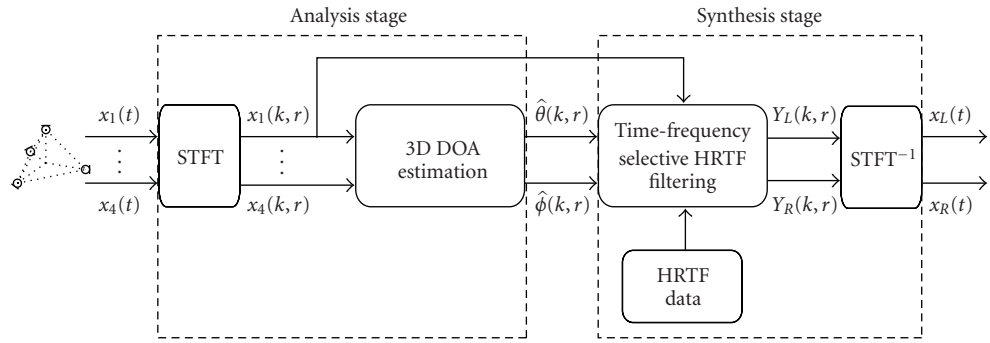


Figure 1: Block diagram of the proposed 3D binaural synthesis method.

and the sound pressure developed at the listener's ear. Since humans have different-sized heads, torsos, and ear shapes, HRTFs vary from person to person. The HRTF is a function of direction, distance, and frequency. The inverse Fourier transform of the HRTF is the *Head-Related Impulse Response* (HRIR), which is a function of direction, distance, and time. Using binaural sound reproduction, it is possible to create a very convincing and immersive sound experience that provides the listener with a natural perception of localized sound events.

In this paper, we present a novel method to capture and process the spatial characteristics of sound with the aim of providing a real-time 3D audio experience. Instead of using an expensive dummy head setup, a small tetrahedral microphone array is utilized to discriminate among the three spatial dimensions, providing a cheap and effective way of constructing a full 3D audio system. The proposed technique is based on a two-step approach. In a first analysis stage, the signals captured by each microphone pair are processed in the time-frequency domain, resulting in a complete directional description of the recorded sound. In the synthesis stage, source sparseness is assumed, and each time-frequency bin is selectively filtered using a different HRTF depending on its estimated direction.

Figure 1 summarizes the steps involved in the proposed approach for binaural sound synthesis.

(1) The signals obtained by the microphones of the array enter the analysis stage.

(2) In the analysis stage, the four signals are first transformed into the time-frequency domain by means of the *Short-Time Fourier Transform* (STFT). Then, *Direction-of-Arrival* (DOA) information (azimuth and elevation) for each time-frequency bin is extracted using the processing described in Section 3.

(3) The synthesis stage is based on a time-frequency selective HRTF filtering of one of the input microphone signals. This filtering is carried out selectively in the STFT domain according to the directions estimated in the analysis stage, resulting in the output signals for the left and right ears. Finally, the ear signals are transformed back to the time domain using the inverse STFT operator.

The paper is structured as follows. Section 2 provides a review of multisource binaural synthesis techniques closely related to our work. Section 3 presents the processing techniques involved in the analysis stage of the method, describing the signal model and the array geometry used to estimate the directional information. Section 4 is devoted to the synthesis stage of the method, where the analyzed spatial information is used to create an immersive 3D sound scene. Section 5 presents a performance comparison between conventional binaural reproduction and our sparsity-based approach using synthetic mixtures of speech and music sources. Section 6 describes and evaluates a real-time prototype that implements the processing described in this paper. Finally, in Section 7, the conclusions of this work are summarized.

2. Binaural Sound Synthesis

2.1. Multisource Binaural Sound Synthesis. It is widely known that binaural sound synthesis is a technique capable of reproducing a virtual sound image of a recorded sound signal at an apparent position in the three-dimensional space. Binaural synthesis is based on the use of HRTFs (or their HRIRs time-domain representation) to filter the audio streams corresponding to different sound sources located at different spatial positions, creating a highly immersive audio experience. As a result, to render N sound sources positioned at N different locations it is necessary to use $2N$ filters (N for the left ear and N for the right ear). The computational complexity is therefore very dependent on the number of sound sources, which makes the real-time rendering of multiple sound sources a very intensive computational task [4, 5]. In this context, many approaches have been proposed to reduce the complexity of multisource binaural synthesis, many of them based on *parametric HRTFs* [6]. Experiments with parametric HRTFs have confirmed that subjects cannot discriminate the parametric HRTF versions from the original ones if a suitable set of parameters are selected within each critical band [7]. Breebaart et al. [8] proposed some methods to provide a multichannel audio experience over stereo headphones from a mixdown of sound source signals and a parametric representation (spatial parameters) of the multichannel original signal in a time-frequency domain. The binaural synthesis stage combines the spatial parameters

of the multichannel signal with the HRTF parameters that describe the virtual loudspeaker setup, resulting in a set of combined binaural parameters that are later used to modify the downmix signal. These rendering methods provide high-quality binaural reproduction of multichannel audio and can be easily combined with multichannel audio coders such as MPEG surround.

Despite being powerful and promising, the above approaches are substantially different from the application covered in this paper. The reason is that they are mainly based on a time-frequency analysis of different loudspeaker signals whereas our proposed method takes as input the signals from a small microphone array, which are successfully employed to describe the sound field in one point of the three-dimensional space. Therefore, the proposed method shares more similarities with another spatial sound processing technique known as *Directional Audio Coding*.

2.2. Directional Audio Coding. Directional Audio Coding (DirAC) is a recently proposed method for spatial sound recording and reproduction [9] which shares many similarities with the binaural synthesis technique described in this paper. In a first analysis stage, DirAC uses typically a B-format microphone to capture the spatial properties of the sound recorded in a given environment (although other alternatives can also be used [10]). In a second stage, the analyzed spatial features are employed to reproduce the recorded sound again by means of an arbitrary loudspeaker setup. Note that although B-format signals are used, there are substantial differences with conventional Ambisonics reproduction [11].

More recently, a binaural synthesis version of DirAC has been proposed to provide spatial sound reproduction over headphones using the conventional DirAC scheme [12]. The main features of this version and their relation to our proposed approach are next discussed.

2.2.1. DirAC Analysis and Synthesis. The analysis stage of DirAC is based on a time-frequency processing of the B-format input signals to estimate direction and diffuseness parameters. To this end, an energetic analysis based on pressure and velocity signals is carried out, which needs for an adequate calibration before starting the processing [13]. Besides using a B-format microphone, different array structures can be employed in this analysis stage with the aim of estimating the necessary direction and diffuseness parameters.

Regarding DirAC synthesis, several alternatives have also been proposed. In the low-bit-rate version, only one omnidirectional signal is transmitted along with the spatial metadata, which is used as the signal that is processed and applied to all the reproduction loudspeakers. Another version uses B-format signals to construct a set of virtual microphone signals that are similarly processed using the metadata obtained from the analysis stage [9].

The transmitted signals are divided into two different streams: the diffuse and the nondiffuse sound stream. The nondiffuse sound is assumed to be the part of sound that has a clear direction and is reproduced by the loudspeaker setup using vector base amplitude panning (VBAP) [14]. In contrast, the diffuse sound stream is assumed to surround the listener and the input signal is decorrelated and played from multiple loudspeakers.

The binaural version of DirAC follows a philosophy similar to that of Breebaart's work in that a virtual loudspeaker setup is assumed and implemented by means of HRTF data. Both diffuse and nondiffuse sound streams are processed in the same way as in the real loudspeaker version but using virtual loudspeakers simulated by means of HRTFs [15].

2.2.2. Relation to the Proposed Approach. As previously commented, DirAC shares many similarities with the binaural synthesis method proposed in this paper, which is also based on a two-step approach. However, substantial differences can be found both in the analysis and the synthesis stages of the algorithm.

As will be seen in Section 3.3, amplitude calibration is not necessary in our proposed analysis stage, since DOA estimation is based only on phase information. Although different microphone array alternatives have already been proposed for DOA estimation in a DirAC context, they either are limited to DOA estimation in the horizontal plane [16] or they use more than 4 microphones [17, 18]. Moreover, as will be later explained, diffuseness is not directly estimated in our proposed approach since the synthesis stage does not rely on this parameter.

On the other hand, the synthesis stage does not assume a virtual loudspeaker setup nor makes a different treatment between diffuse and nondiffuse components. This makes the synthesis processing even more simple than in DirAC. In fact, in our method, diffuseness information is assumed to be inherently encoded by the DOA estimates since the variance found on the directional information over the time-frequency domain is already a representation of the diffuseness characteristics of the recorded sound. In this context, there is no need for assuming a specific loudspeaker reproduction setup since each time-frequency element is binaurally reproduced according to its estimated direction.

3. Analysis Stage

3.1. Signal Model. The signals recorded by a microphone array, with sensors denoted with indices $m = 1, 2, \ldots, M$ in an acoustic environment where N sound sources are present, can be modeled as a finite impulse response convolutive mixture, written as

$$x_m(t) = \sum_{n=1}^{N} \sum_{\ell=0}^{L_m-1} h_{mn}(\ell) s_n(t-\ell), \quad m = 1, \ldots, M, \qquad (1)$$

where $x_m(t)$ is the signal recorded at the mth microphone at time sample t, $s_n(t)$ is the nth source signal, $h_{mn}(t)$ is the impulse response of the acoustic path from source n to sensor m, and L_m is the maximum length of all impulse responses.

The above model can also be expressed in the STFT domain. This transform divides a time domain signal into a series of small overlapping pieces; each of these pieces is

windowed and then individually Fourier transformed [19]. Using this transform, the model of (1) can be expressed as

$$X_m(k,r) = \sum_{n=1}^{N} H_{mn}(k)S(k,r), \tag{2}$$

where $X_m(k,r)$ denotes the STFT of the mth microphone signal, being k and r the frequency index and time frame index, respectively. $S_n(k,r)$ denotes the STFT of the source signal $s_n(t)$ and $H_{mn}(k)$ is the frequency response from source n to sensor m. Note that (2) is only equivalent to (1) in the case when the analysis window in the computation of the STFT is longer than L_m.

If we assume that the sources rarely overlap at each time-frequency point, (2) can be simplified as follows:

$$X_m(k,r) \approx H_{ma}(k)S_a(k,r), \tag{3}$$

where $S_a(k,r)$ is the dominant source at time-frequency point (k,r). To simplify, we assume an anechoic model where the sources are sufficiently distant to consider plane wavefront incidence. Then, the frequency response is only a function of the time-delay τ_{mn} between each source and sensor

$$H_{mn}(k) = e^{j2\pi f_k \tau_{mn}}, \tag{4}$$

f_k being the frequency corresponding to frequency index k.

3.2. Sparsity and Disjointness. Speech and music signals have been shown to be sparse in the time-frequency domain [20]. A sparse source has a peaky probability density function; the signal is close to zero at most time-frequency points, and has large values in rare occasions. This property has been widely applied in many works related to source signal localization [21, 22] and separation [23, 24] in underdetermined situations, that is, when there are more sources than microphone signals. However, source sparsity alone is useless if the sources overlap to a high degree. The *disjointness* of a mixture of sources can be defined as the degree of nonoverlapping of the mixed signals. An objective measure of disjointness is the so-called *W-Disjoint Orthogonality* (WDO) [25, 26].

Spectral overlapping depends not only on source sparsity, but also on the mutual relationships between signals. Highly uncorrelated signals will result in a low probability of overlapping. This is even truer for statistically independent signals, since independence is a stronger requirement than uncorrelation. Speech signals most often mix in a random and uncorrelated manner, such as in the cocktail party paradigm. With music mixtures, the situation is different. Their disjointness will vary strongly according to music type. Tonal music will result in strong overlaps in frequency, while atonal music will be more disjoint in frequency [27].

The disjointness properties of speech and music signals are dependent on the window size parameter, which affects the number of frequency bands in the analysis. In particular, the disjointness of speech signals decreases when the window size is very large as a consequence of the reduced temporal resolution. For music signals, frequency disjointness plays a more important role than time disjointness and so frequency resolution should be favored with longer analysis windows. Moreover, as expected, mixtures of correlated melodies shown to be less disjoint than uncorrelated ones due to the higher amount of spectral and temporal overlapping.

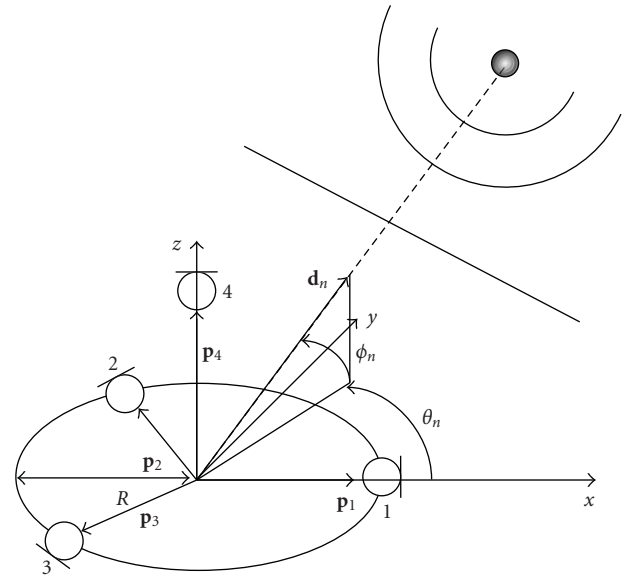


Figure 2: Tetrahedral microphone array for 3D DOA estimation.

It is also worth to remark that the sparsity and disjointness properties of audio signals become affected in reverberant environments. The room impulse response smears the energy in both time and frequency and so the spectral overlap between different sources in the time-frequency domain is increased with reverberation. Despite this effect, the assumption of nonoverlapping sources has been shown to be still useful for sparsity-based applications such as source separation [28, 29].

3.3. Array Geometry and DOA Estimation. Now consider a tetrahedral microphone array ($M = 4$) with base radius R, as shown in Figure 2. The sensor location vectors in the 3-dimensional space with origin in the array base center, are given by

$$\begin{aligned}
\mathbf{p}_1 &= [R,\ 0,\ 0]^T,\\
\mathbf{p}_2 &= \left[-\frac{R}{2},\ \frac{\sqrt{3}}{2}R,\ 0\right]^T,\\
\mathbf{p}_3 &= \left[-\frac{R}{2},\ -\frac{\sqrt{3}}{2}R,\ 0\right]^T,\\
\mathbf{p}_4 &= \left[0,\ 0,\ R\sqrt{2}\right]^T.
\end{aligned} \tag{5}$$

The DOA vector of the nth source as a function of the azimuth θ_n and elevation ϕ_n angles is defined as

$$\mathbf{d}_n = [\cos\theta_n \cos\phi_n, \sin\theta_n \cos\phi_n, \sin\phi_n]^T. \tag{6}$$

The source to sensor time delay with respect to the origin is given by $\tau_{mn} = \mathbf{p}_m^T \mathbf{d}_n / c$, c being the speed of sound. Therefore, the frequency response of (4) can be written as

$$H_{mn}(k,r) \approx e^{\mathrm{j}(2\pi f_k/c)\mathbf{p}_m^T \mathbf{d}_n}. \tag{7}$$

Taking into account this last result and (3), it becomes clear that the phase difference between the microphone pair formed by sensors i and j, is given by

$$\angle\left(\frac{X_j(k,r)}{X_i(k,r)}\right) \approx \frac{2\pi f_k}{c}\left(\mathbf{p}_j - \mathbf{p}_i\right)^T \mathbf{d}_n, \tag{8}$$

where $\angle$ denotes the phase of a complex number.

Using a reference microphone q, the phase difference information at point (k,r) of $M-1$ microphone pairs is stored in the vector

$$\mathbf{b}_q(k,r) = \left[\angle\left(\frac{X_1(k,r)}{X_q(k,r)}\right), \ldots, \angle\left(\frac{X_M(k,r)}{X_q(k,r)}\right)\right]^T, \tag{9}$$

forming the following system of equations:

$$\mathbf{b}_q(k,r) = \frac{2\pi f_k}{c}\mathbf{P}\mathbf{d}_n, \tag{10}$$

where

$$\mathbf{P} = \left[\mathbf{p}_{1q}, \ldots, \mathbf{p}_{Mq}\right]^T, \qquad \mathbf{p}_{nq} = \mathbf{p}_n - \mathbf{p}_q. \tag{11}$$

Finally, the DOA at time-frequency bin (k,r) is obtained by taking the inverse of the $\mathbf{P}$ matrix

$$\hat{\mathbf{d}}_n(k,r) = \frac{c}{2\pi f_k}\mathbf{P}^{-1}\mathbf{b}_q(k,r). \tag{12}$$

The regular tetrahedral geometry used in this paper leads to the following simple equations for $\mathbf{d}_n(k,r) = [\hat{d}_1,\ \hat{d}_2,\ \hat{d}_3]^T$:

$$\begin{aligned}
\hat{d}_1 &= \cos\theta_n \cos\phi_n = \frac{c}{2\pi f_k}\frac{1}{\sqrt{3}}(b_2 + b_3),\\
\hat{d}_2 &= \sin\theta_n \cos\phi_n = \frac{c}{2\pi f_k}(b_3 - b_2),\\
\hat{d}_3 &= \sin\phi_n = \frac{c}{2\pi f_k}\left[\frac{1}{\sqrt{6}}(b_2 + b_3) - \sqrt{\frac{3}{2}}b_4\right],
\end{aligned} \tag{13}$$

where b_n is the nth element of the vector $\mathbf{b}_1(k,r)$ (reference microphone $q = 1$). The azimuth angle is obtained using the four quadrant inverse tangent function:

$$\hat{\theta}_n(k,r) = \mathrm{atan}^{360^\circ}\left(\hat{d}_1, \hat{d}_2\right). \tag{14}$$

The elevation angle is directly obtained as

$$\hat{\phi}_n(k,r) = \sin^{-1}\left(\hat{d}_3\right). \tag{15}$$

Note that for each time-frequency point (k,r), estimating the 3D direction of arrival is relatively simple, just using the observed phase differences between 3 microphone pairs of the array. Another aspect to consider is spatial aliasing. The distance between microphones determines the angular aliasing frequency. Due to the 2π ambiguity in the calculation of the phase differences, the maximum ambiguity-free frequency in a microphone pair subarray would be given by $f_k = c/2d$, where d is the separation distance between the capsules. Beyond this frequency, there is not a one-to-one relationship between phase difference and spatial direction. However, small arrays with $d \approx 1.5$ cm provide an unambiguous bandwidth greater than 11 kHz, covering a perceptually important frequency range.

3.4. Example. With the objective of showing how this analysis stage is capable of capturing the 3D spatial information of sound, we show a simulated sound scene where 4 speech sources are simultaneously active. The simulation has been carried out considering a shoe-box-shaped room (3.6 m × 3.6 m × 2.2 m) with reflecting walls (reverberation time $T_{60} = 0.1$ s). The azimuth angles of the sources were $\theta_1 = 15^\circ$, $\theta_2 = 75^\circ$, $\theta_3 = 210^\circ$, and $\theta_4 = 260^\circ$. The elevation angles were $\phi_1 = 0^\circ$, $\phi_2 = 30^\circ$, $\phi_3 = -10^\circ$, and $\phi_4 = 45^\circ$. Figure 3(a) shows the source locations in the 3D space. Figures 3(b) and 3(c) show the 2D histograms of the distribution of DOA estimates in the XY and ZY plane, where red color means that many estimates are concentrated on the same location. Note how most DOA estimates are concentrated around the actual source directions. The deviations in the estimates are a consequence of room reflections and interference. The effect of reverberation in sparse source localization was studied by the authors in a previous work [30]. However, as will be explained in the next section, these deviations do not have a negative effect on our proposed binaural synthesis method, since they contribute to the perception of the diffuseness properties of sound.

4. Synthesis Stage

As said in Section 1, HRTFs provide accurate localization cues because they encode all the necessary information regarding how the arriving sound is filtered by the diffraction and reflection properties of the head, pinna, and torso, before it reaches the eardrum and inner ear. Using this information, synthesizing a binaural sound signal for headphone reproduction is straightforward. The HRTF for each ear must be used to filter an input signal, ensuring that the outputs are correctly reproduced over their corresponding headphone channel. This is usually done for each separate source signal with the aim of positioning an auditory object in the direction from which the HRTFs have been measured. However, in our proposed approach, the synthesis stage differs significantly from this conventional processing due to the fact that no separate source signals are available. Again, taking into account source sparseness in the time-frequency domain, we will be able to reproduce the original spatial characteristics of the recorded sound scene using the directional information extracted from the previous analysis stage.

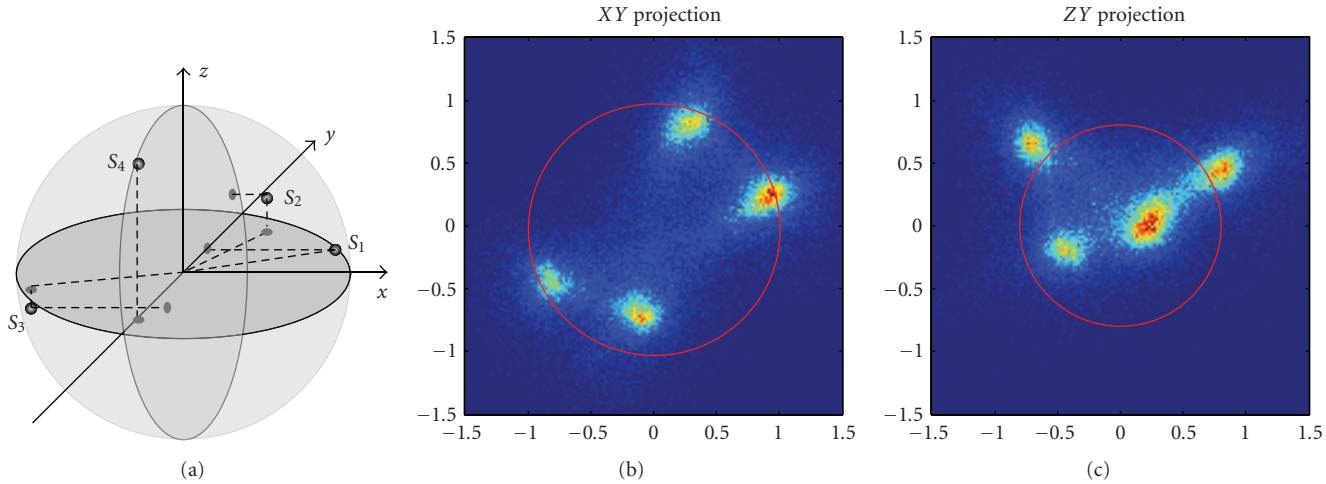


Figure 3: DOA analysis of a mixture of 4 speech sources. (a) Source locations in the 3D space. (b) Distribution of DOA estimates in the XY plane. (c) Distribution of DOA estimates in the ZY plane.

4.1. Time-Frequency Selective HRTF Filtering. Consider a set of measured [31, 32] or simulated HRTFs [33]. It is widely known that the use of nonindividualized HRTFs for binaural reproduction has some problems, mainly

(i) sound objects are frequently localized inside the head,

(ii) frontal sounds often appear behind the listener and vice versa,

(iii) the perceived directions of the synthesized sources do not match the intended spatial positions.

These classical problems associated to binaural reproduction have already been extensively studied [34] and we will not address them in this paper.

Assuming far field conditions, the HRTF is a function of the arrival direction of the source (θ_n, ϕ_n) and the frequency f_k, expressed as $\mathrm{HRTF}(\theta_n, \phi_n, k)$. Moreover, there is also a different HRTF for the right and left ears, having $\mathrm{HRTF}_L(\theta_n, \phi_n, k)$ and $\mathrm{HRTF}_R(\theta_n, \phi_n, k)$.

The synthesis strategy is simple. Any of the omnidirectional signals of the array $X_m(k, r)$ is filtered accordingly to the estimated DOA angles $\hat{\theta}_n$ and $\hat{\phi}_n$ as follows:

$$\begin{aligned} Y_L(k, r) &= X_m(k, r)\mathrm{HRTF}_L\left(\hat{\theta}_n, \hat{\phi}_n, k\right), \\ Y_R(k, r) &= X_m(k, r)\mathrm{HRTF}_R\left(\hat{\theta}_n, \hat{\phi}_n, k\right), \end{aligned} \tag{16}$$

where $Y_L(k, r)$ and $Y_R(k, r)$ are the STFT of the output signals corresponding to the left and right ears, respectively. These signals are transformed back to the time domain using the inverse STFT operator following an overlap-add scheme.

Using the above approach, the microphone signal $X_m(k, r)$ provides a pressure signal for the synthesis of the binaural signal. The required spatial localization cues are then given by the HRTF coefficients, which are carefully selected based on the estimated directional data. Note that we only use a single omnidirectional signal for the calculation of the output, since combinations of the microphone signals could result in coloration due to spatial filtering effects. In our implementation, we chose the signal of microphone 4 for being slightly above from the array center.

Table 1: Mean square error for synthesized signals in anechoic scenario.

Number of sources N	Mean square error
1	0.003
2	0.073
3	0.243
4	0.382

4.2. Selective Filtering and Sparsity. Further considerations are needed regarding the above synthesis approach. Note that each time-frequency bin is independently filtered according to its DOA information. As well as in the analysis stage, source sparsity and disjointness form the basis of our synthesis method. Under approximate WDO conditions, only one source has a major contribution on a given time-frequency element. Therefore, it is only necessary to filter each bin according to the direction of the dominant contribution since the energy corresponding to the rest of the sources can be neglected with little error. Obviously, if the number of sources is increased, the error will be higher.

To illustrate this idea, Figure 4 shows the waveforms of the left and right ear signals in an anechoic scenario for an increasing number of sources. The real signals obtained by conventional HRTF synthesis are shown on the left side and the ones synthesized by means of time-frequency selective filtering are on the right side. To evaluate quantitatively the synthesized signals, their mean square errors are provided in Table 1. As expected, the error of the synthesized signal depends on the number of sources. However, as will be shown in the next section, these errors do not severely affect the subjective quality of the synthesized signals.

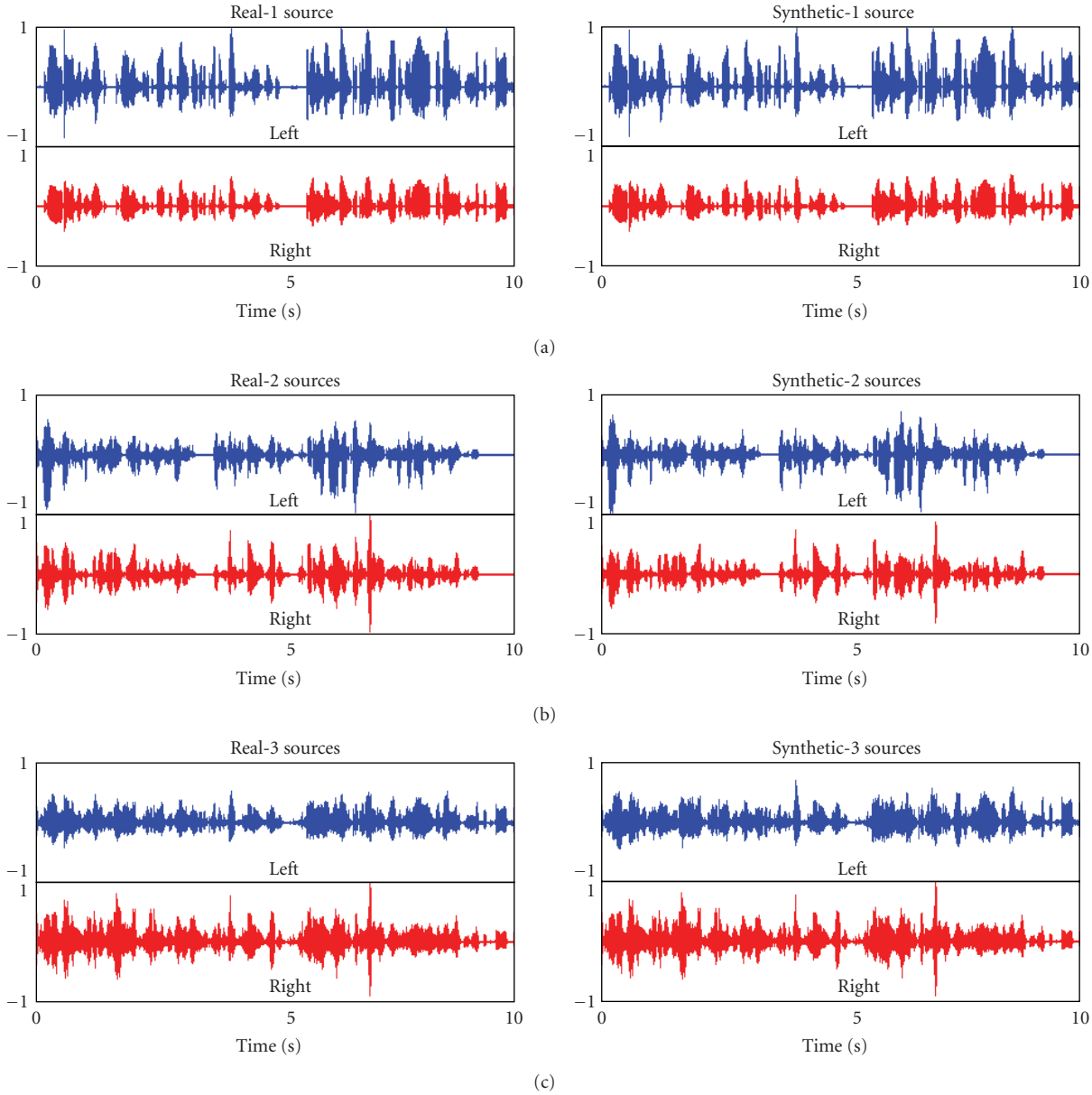


Figure 4: Waveforms of real and sparsity-based synthesized binaural signals for different number of sources. (a) One source. (b) Two sources. (c) Three sources.

Besides the number of sources, the environment and the type of signals also play a major role in our synthesis method. As explained in Section 3.2, speech and music have different sparsity properties and room reflections spread the energy of the sources both in time and frequency. However this is not a serious problem. Time-frequency points dominated by reverberation will be inherently reproduced from multiple directions, just as suggested by the analysis stage. This way, the higher the variance found in the estimated directions, the higher the sense of envelopment will be perceived. In contrast, a dry room that produces very peaky DOA distributions will result in a synthesized binaural signal where the sources can be clearly perceived from their actual directions. A problem associated with a high degree of reverberation is that artifacts may appear due to the prominent discontinuities in the directional data. These effects can be effectively reduced by smoothing the filtering coefficients along the frequency axis.

4.3. HRTF Spatial Resolution. Traditionally, a practical problem associated with HRTFs is the difficulty to measure responses for every possible angle with infinite spatial resolution. Although some approaches have been recently proposed to solve this classical problem [35], most available HRTF databases have been measured with some practical resolution. In order to use HRTFs corresponding to the acquired directional information, several approaches can be followed.

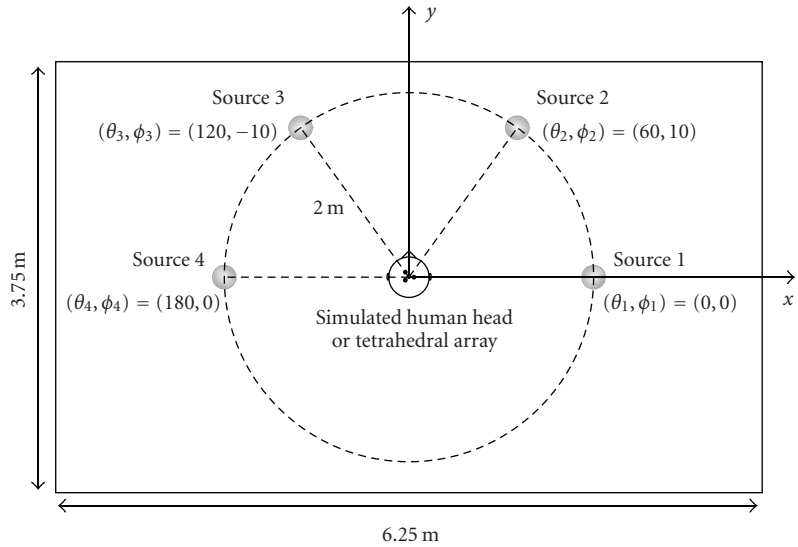


Figure 5: Simulation setup used to obtain the signals used in the subjective evaluation.

(1) Use directly the HRTF of the available data bank that is closest to the estimated direction. This would be the simplest approach.

(2) Interpolate the available HRTFs to get more accurate filters. This can be done using sophisticated interpolation techniques [36–38]. However, a simple linear interpolation in the time domain using the closest HRIRs has shown to be a very convincing and efficient solution [39, 40].

(3) Use a parametric HRTF model [33, 41]. This option provides directly the filtering information needed for any direction.

Depending on the requirements of a given application, a different strategy can be selected. While the interpolation strategy is very useful for achieving accurate localization, the other two methods are computationally more efficient. In the next section we comment on some useful aspects regarding the real-time implementation of the method.

5. Evaluation Using Synthetic Mixtures

To evaluate subjectively the quality and spatial impression provided by the proposed technique, a set of simulations considering different acoustic situations were carried out. The evaluation conducted this way is useful to assess the performance of the method under different acoustic environments with control on specific aspects of the acoustic setup.

In the experiments, a set of sound sources were simulated inside a shoe-box-shaped room ($6.25 \times 3.75 \times 2.5$ m), acquiring all the required impulse responses by means of the *Roomsim* [42] simulation package for Matlab. This software simulates the acoustics of a rectangular room by means of the image method [43] and, moreover, it allows to generate binaural room impulse responses (BRIRs) corresponding to a selected HRTF data set.

The simulation setup is depicted in Figure 5. Four source positions, were considered in the experiments at a radius 2 m for the array base center (origin of coordinates): ($\theta_1 = 0^\circ, \phi_1 = 0^\circ$), ($\theta_2 = 60^\circ, \phi_2 = 10^\circ$), ($\theta_3 = 120^\circ, \phi_3 = -10^\circ$), and ($\theta_4 = 180^\circ, \phi_4 = 0^\circ$). The signals at the microphones were obtained by convolving the simulated responses with the corresponding dry source signals and adding all of them together. To simulate our tetrahedral array, we used an intermicrophone distance of $d = 1.5$ cm and assumed perfect omnidirectional responses for all sensors. On the other hand, the KEMAR mannequin [44] was selected to generate reference source signals for the subjective tests.

Different types of signals were considered to take into account different sparsity properties.

(i) A set of 2 male and 2 female speech sources extracted from the public data provided in *The 2008 Signal Separation Evaluation Campaign* [45]. They are sampled at 16 kHz, 16 bits, and have a duration of 10 s.

(ii) A multitrack folk music recording consisting of four instruments: accordion, sax, guitar, and violin. Although originally sampled at 44.1 kHz, they were resampled to have the same sampling frequency (16 kHz) as the above speech mixtures.

The STFT was computed using Hann windows of 1024 samples of length, with a hop size of 512 samples (50% overlap). These parameters have been shown to be optimum for sparsity-based speech processing [27]. However, music would benefit from longer time windows.

A set of 7 listeners took part on an informal listening test with the aim of evaluating the similarities between the scenes rendered by means of the simulated KEMAR and those obtained by means of the proposed approach. The assessed sound scenes were mixtures of one, two, three, and four sources. There were three versions of each scene. Each version was obtained using different room surface

characteristics, thus, having different reverberation times: $T_{60} = 0$ s (anechoic), $T_{60} = 0.1$ s (slightly reverberant), and $T_{60} = 0.9$ (very reverberant). As a result, there were 2 versions (KEMAR-simulation and proposed) of a total of 24 different sound scenes (12 for speech and 12 for music).

Two different aspects were considered in the evaluation: sound quality and spatial impression. A 4-point grade scale was used to compare the scenes rendered using the tetrahedral array with the reference KEMAR simulated scenes, ranging from −3 to 0 in the following intensity scale:

(i) 0: Equal,

(ii) −1: Slightly Worse,

(iii) −2: Moderately Worse,

(iv) −3: Substantially Worse.

5.1. Results. Figures 6(a) and 6(b) show the results of the tests for sound quality and spatial impression, respectively. Black dots denote the mean values and thin bars represent 95% confidence intervals. Regarding sound quality (Figure 6(a)), it can be observed that in anechoic conditions ($T_{60} = 0$), there are no significant differences between both binaural reproduction methods. However, as the reverberation degree gets higher, the performance of the method is slightly degraded. This worsening may be due to some metallic sound reported by some listeners. There are also clear differences between speech and music, music being considerably more problematic than speech, specially when the number of sound sources is higher. This is a consequence of harmonic overlapping, which affects substantially the WDO assumption. Regarding spatial impression (Figure 6(b)), the decreasing tendency with reverberation is again observed, but the number of sound sources and the type of source signals seem to be less significant.

From the above results, it becomes clear that both source overlapping and reverberation affect negatively the performance of the proposed approach. Obviously, this degradation is due to the fact that some of the assumptions taken for the development of the algorithm are not completely met, specially those based on source sparsity and disjointness. A detailed analysis of the artifacts caused by different types of errors in the analysis and synthesis stages could be useful to improve the performance of the method when working in difficult acoustic environments. Although this analysis is out of the scope of this paper, the authors plan to address this issue in future works.

6. Evaluation with Real Mixtures

6.1. Real-Time Implementation. In the last section, a set of experiments using simulations of reverberant rooms were presented. Besides considering these simulations, the applicability of the proposed method can be substantially enhanced by providing some notes on the real-time implementation of a working prototype. Two objectives are pursued with this implementation. First, to demonstrate that the computational cost of this technique is reduced enough to be implemented in a practical embedded system. Second, having a real-time system allowed us to plan future interactive experiments where conditions related to scene changes can be experienced as they occur.

For our real-time prototype we used a PC running Microsoft Windows XP as a base. To construct the microphone array prototype with $d = 1.5$ cm, four instrumentation quality microphones from Brüel & Kjaer model 4958 were used. These microphones have excellent phase matching in the audio band. The signal acquiring system consisted of a digital audio interface with four microphone inputs (M-Audio Fast Track Ultra USB 2.0) and ASIO drivers. The *Intel Integrated Performance Primitives* (Intel IPP) [46] library was used for FFT computation and vector operations. In the analysis stage, phase differences are calculated from the FFT coefficients of each input data frame at each channel. The $[x, y, z]$ components of the DOA vector are then calculated using (13), taking into account that the corresponding frequencies f_k have to be previously stored. Moreover, the processing parameters were set the same as in Section 5.

Since the experiments reported in the following subsection were conducted using a Brüel & Kjaer 4128 *Head And Torso Simulator* (HATS), the HRTF database used for the synthesis was specifically measured by the authors to allow for an objective comparison. The HRTFs were measured using the logarithmic sweep method [47] with sampling frequency 44.1 kHz. Moreover, the measuring system was carefully compensated. HRTFs were sampled both in azimuth and elevation. The dummy-head was placed in a rotating table, measuring responses from −180° to 180° every 5 degrees. On the other hand, elevations were measured from −40° to 90° every 10 degrees. For every measure, the same loudspeaker distance to the center was employed (1 m).

6.2. Evaluation and Discussion. Experiments similar to those presented in Section 5 were carried out using the constructed prototype. Different combinations of sound sources were simultaneously recorded using the tetrahedral microphone array and the HATS, placing the microphone array on top (Figure 7). The sources were reproduced in the horizontal plane over different loudspeakers of our Wave-Field Synthesis array, with azimuth angles of 0, 60, 120, and 180 degrees. The room has an approximate reverberation time of $T_{60} = 0.2$ s. For comparison purposes, the same speech and music signals used in the simulations were selected. The reference signals for the listening test are the simultaneously recorded signals from the artificial head. The same group of subjects took part in the evaluation.

The results of this experiment are shown in Figure 8. As expected, there are many similarities with those in Figure 6 for slight reverberation. Again, results both in sound quality and spatial impression are worse for music signals than for speech signals, specially when the number of sources is high. Moreover, the results confirm that sound quality is more critical than spatial impression, however, the overall score suggests that the perceived quality obtained with the

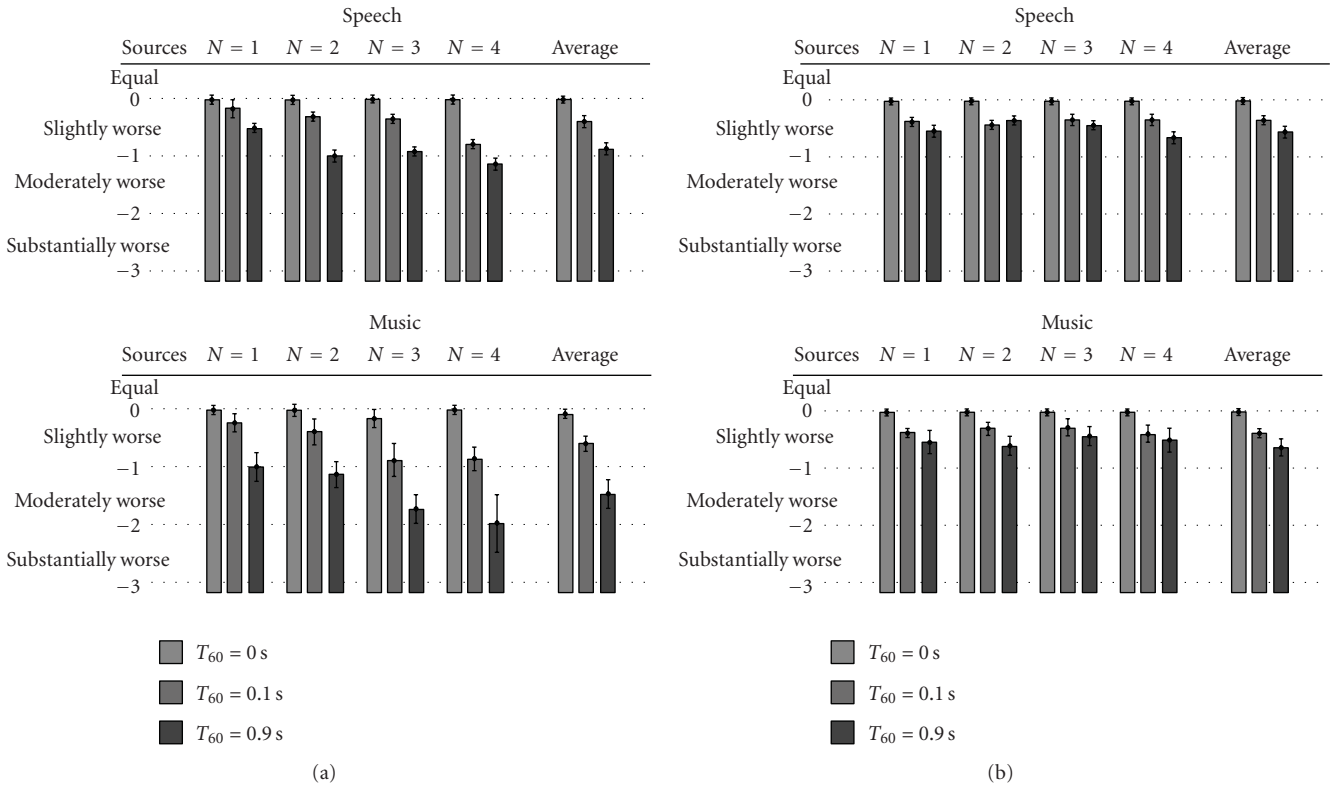


Figure 6: Results of the subjective tests using synthetic mixtures. Black dots denote the mean values and thin bars represent 95% confidence intervals. (a) Sound quality evaluation. (b) Spatial impression evaluation.

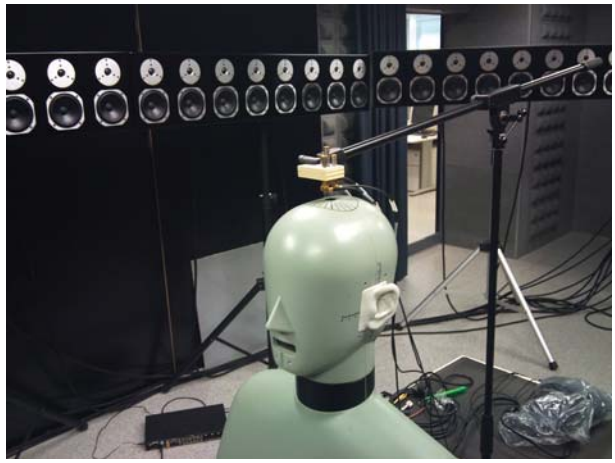


Figure 7: Tetrahedral array and acoustic dummy-head used in the experiments.

proposed synthesis method is only slightly degraded from the obtained using the acoustic dummy-head.

7. Conclusion

In this paper, we have presented a two-step binaural sound synthesis method based on sparse signal processing and time-frequency analysis. In the first stage, the assumption of sound sources that rarely overlap in the time-frequency domain has been considered to study the spatial properties of the sound that impinges a small tetrahedral microphone array. The phase difference information of several microphone pairs is combined to obtain a 3D DOA estimate in each time-frequency slot. In the synthesis stage, one of

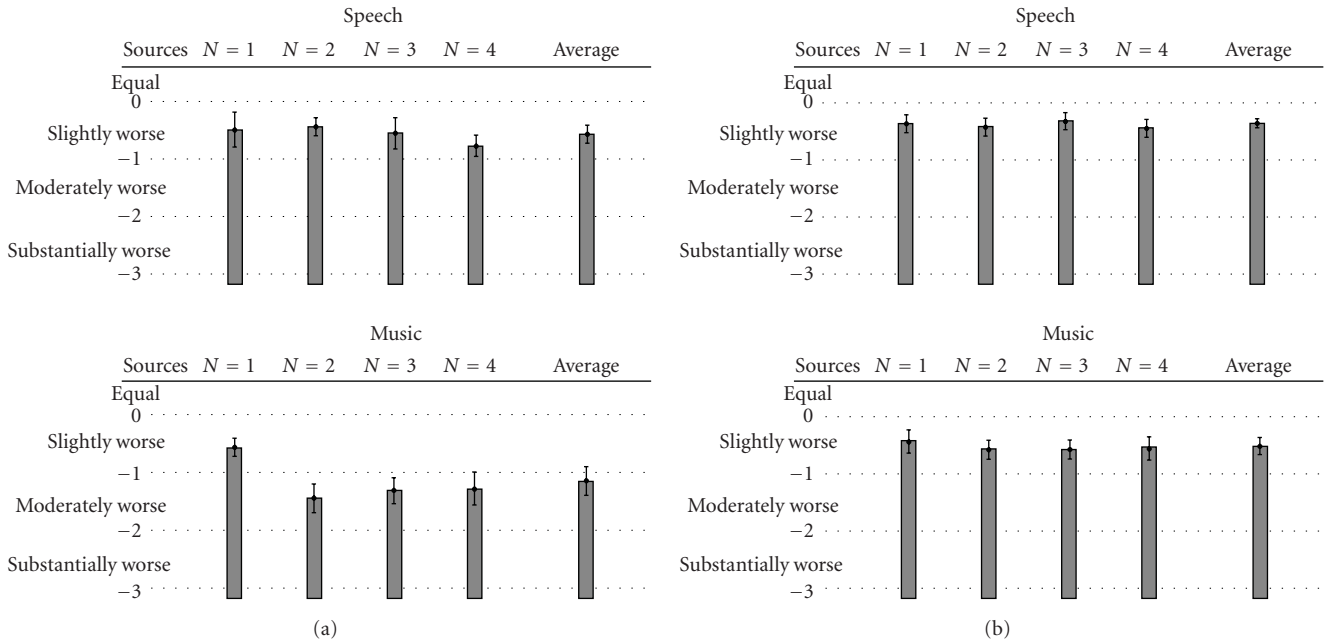


Figure 8: Results of the subjective tests using real recorded mixtures. Black dots denote the mean values and thin bars represent 95% confidence intervals. (a) Sound quality evaluation. (b) Spatial impression evaluation.

the microphone signals is selectively filtered in the time-frequency domain with the left and right HRTFs that correspond to the estimated DOAs.

Experiments using both synthetic and real mixtures of speech and music were conducted using different number of sources. Although the performance of the method is slightly degraded with the number of sources and reverberation, the perceived sound quality and spatial impression are considerably similar to conventional binaural reproduction. However, artifacts due to spectral overlapping makes this method more suitable for speech applications than for music.

The proposed spatial sound capturing method not only eliminates the need for an acoustic mannequin, which has a considerable volume and uncomfortable portability, but also allows to change easily the head response by using a different HRTF database in requirement of the application, needs, or user preferences. Moreover, it allows to rotate the head position in real time. Thus, a tracking system can be used to follow the position of the subject in the synthesis stage, providing the listener with a more immersive sensation.

Acknowledgment

The Spanish Ministry of Science and Innovation supported this work under the Project TEC2009-14414-C03-01.

References

[1] J. Blauert, *Spatial Hearing*, MIT Press, Cambridge, UK, 1997.

[2] F. Rumsey, *Spatial Audio*, Focal Press, 2001.

[3] C. I. Cheng and G. H. Wakefield, "Introduction to head-related transfer functions (HRTF's): representations of HRTF's in time, frequency, and space," in *Proceedings of the 107th Convention of the Audio Engineering Society (AES '99)*, New York, NY, USA, 1999.

[4] P. S. Chanda, S. Park, and T. I. Kang, "A binaural synthesis with multiple sound sources based on spatial features of head-related transfer functions," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1726–1730, Vacouver, Canada, July 2006.

[5] P. G. Georgiou and C. Kyriakakis, "A multiple input single output model for rendering virtual sound sources in real time," in *Proceedings of the IEEE Internatinal Conference on Multimedia and Expo (ICME '00)*, vol. 1, pp. 253–256, New York, NY, USA, July 2000.

[6] J. Breebaart, F. Nater, and A. Kohlrausch, "Parametric binaural synthesis: background, applications and standards," in *Proceedings of the NAG-DAGA*, pp. 172–175, Rotterdam, The Netherlands, 2009.

[7] J. Breebaart and C. Faller, *Spatial Audio Processing: MPEG Surround and Other Applications*, Wiley, Chichester, UK, 2007.

[8] J. Breebaart, L. Villemoes, and K. Kjörling, "Binaural rendering in MPEG surround," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 732895, 14 pages, 2008.

[9] V. Pulkki, "Spatial sound reproduction with directional audio coding," *Journal of the Audio Engineering Society*, vol. 55, no. 6, pp. 503–516, 2007.

[10] G. Del Galdo and F. Kuech, "Nested microphone array processing for parameter estimation in directional audio coding," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '09)*, New Paltz, NY, USA, October 2009.

[11] V. Pulkki, "Directional audio coding in spatial sound reproduction and stereo upmixing," in *Proceedings of the AES 28th International Conference*, Pitea, Sweden, July 2006.

[12] M. Laitinen and V. Pulkki, "Binaural reproduction for directional audio coding," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '09)*, pp. 337–340, New Paltz, NY, USA, October 2009.

[13] O. Thiergart, G. Del Galdo, M. Taseska, J. Pineda, and F. Kuech, "In situ microphone array calibration for parameter estimation in directional audio coding," in *Proceedings of the AES 128th Convention*, London, UK, May 2010.

[14] V. Pulkki, "Spatial sound generation and perception by amplitude panning techniques," Tech. Rep., Helsinki University of Technology, Helsinki, Finland, 2001.

[15] M. Laitinen, *Binaural reproduction for directional audio coding*, M.S. thesis, Helsinki University of Technology, Helsinki, Finland, 2008.

[16] J. Ahonen, V. Pulkki, and T. Lokki, "Teleconference application and B-format microphone array for directional audio coding," in *Proceedings of the AES 30th International Conference*, Saariselkä, Finland, March 2007.

[17] R. Schultz-Amling, F. Kuech, M. Kallinger, G. Del Galdo, J. Ahonen, and V. Pulkki, "Planar microphone array processing for the analysis and reproduction of spatial audio using directional audio coding," in *Proceedings of the AES 124th Convention*, Amsterdam, The Netherlands, May 2008.

[18] J. Merimaa, "Applications of a 3-d microphone array," in *Proceedings of the AES 112th Convention*, Munich, Germany, May 2002.

[19] L. Cohen, *Time-Frequency Analysis*, Prentice-Hall, 1995.

[20] J. J. Burred and T. Sikora, "On the use of auditory representations for sparsity-based sound source separation," in *Proceedings of the 5th International Conference on Information, Communications and Signal Processing (ICICS '05)*, pp. 1466–1470, Bangkok, Thailand, December 2005.

[21] S. Araki, H. Sawada, R. Mukai, and S. Makino, "Performance evaluation of sparse source separation and DOA estimation with observation vector clustering in reverberant environments," in *Proceedings of the International Workshop on Acoustic Echo and Noise Control(IWAENC '06)*, Paris, France, 2006.

[22] S. Rickard and F. Dietrich, "DOA estimation of many w-disjoint orthogonal sources from two mixtures using DUET," in *Proceedings of the 10th IEEE Workshop on Statistical Signal and Array Processing (SSAP '00)*, pp. 311–314, Pocono Manor, Pa, USA, August 2000.

[23] Ö. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[24] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Processing*, vol. 81, no. 11, pp. 2353–2362, 2001.

[25] S. Rickard and O. Yilmaz, "On the w-disjoint orthogonality of speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 529–532, Orlando, Fla, USA, May 2002.

[26] A. Jourjine, S. Richard, and O. Yilmaz, "Blind separation of disjoint orthogonal signals: demixing n sources from 2 mixtures," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '00)*, vol. 5, pp. 2985–2988, Istanbul, Turkey, 2000.

[27] J. J. Burred, *From sparse models to timbre learning: new methods for musical source separation*, Ph.D. thesis, Technical University of Berlin, 2008.

[28] S. Schulz and T. Herfet, "On the window-disjoint-orthogonality of speech sources in reverberant humanoid scenarios," in *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx '08)*, Espoo, Finland, September 2008.

[29] M. Cobos and J. J. Lopez, "Two-microphone separation of speech mixtures based on interclass variance maximization," *Journal of the Acoustical Society of America*, vol. 127, no. 3, pp. 1661–1672, 2010.

[30] M. Cobos, J. J. Lopez, and S. Spors, "Effects of room reverberation in source localization using small microphone arrays," in *Proceedings of the 4th International Symposium on Communications, Control and Signal Processing (ISCCSP '10)*, Limassol, Cyprus, March 2010.

[31] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF database," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 99–102, New Paltz, NY, USA, October 2001.

[32] IRCAM, "LISTEN HRTF database," 2003, http://recherche ircam.fr/equipes/salles/listen/.

[33] C. P. Brown and R. O. Duda, "An efficient HRTF model for 3-D sound," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (ASSP '97)*, 1997.

[34] H. Møller, M. F. Sørensen, C. B. Jensen, and D. Hammershøi, "Binaural technique: do we need individual recordings?" *Journal of the Audio Engineering Society*, vol. 44, no. 6, pp. 451–468, 1996.

[35] G. Enzner, "3-d-continuous-azimuth acquisition of head-related impulse responses using multi-channel adaptive filtering," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '09)*, pp. 325–328, New Paltz, NY, USA, October 2009.

[36] F. Keyrouz and K. Diepold, "Efficient state-space rational interpolation of HRTFs," in *Proceedings of the AES 28th International Conference*, Pitea, Sweden, 2006.

[37] F. Freeland, L. Biscainho, and P. Diniz, "Efficient HRTF interpolation in 3D moving sound," in *Proceedings of the AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, June 2002.

[38] F. Keyrouz and K. Diepold, "A new HRTF interpolation approach for fast synthesis of dynamic environmental interaction," *Journal of the Audio Engineering Society*, vol. 56, no. 1-2, pp. 28–35, 2008.

[39] J. Sodnik, R. Sušnik, M. Štular, and S. Tomažič, "Spatial sound resolution of an interpolated HRIR library," *Applied Acoustics*, vol. 66, no. 11, pp. 1219–1234, 2005.

[40] T. Nishino, S. Mase, S. Kajita, K. Takeda, and F. Itakura, "Interpolating HRTF for auditory virtual reality," *Journal of the AcousticalSociety of America*, vol. 100, no. 4, pp. 2602–2602, 1996.

[41] V. Algazi, R. Duda, and D. M. Thomson, "The use of head-and-torso models for improved spatial sound synthesis," in *Proceedings of the AES 113th Convention*, Los Angeles, Calif, USA, October 2002.

[42] D. R. Campbell, "Roomsim: a MATLAB simulation shoebox room acoustics," 2007, http://media.paisley.ac.uk/~campbell/Roomsim/.

[43] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society ofAmerica*, vol. 65, no. 4, pp. 943–950, 1979.

[44] B. Gardner and K. Martin, "HRTF measurements of a KEMAR dummy-head microphone," MIT Media Lab, May 1994, http://alumni.media.mit.edu/~kdm/hrtfdoc/hrtfdoc.html.

[45] E. Vincent, S. Araki, and P. Bofill, "The 2008 signal separation evaluation campaign: a community-based approach to large-scale evaluation," in *Independent Component Analysis and Signal Separation*, vol. 5441 of *Lecture Notes in Computer Science*, pp. 734–741, 2009.

[46] S. Taylor, *Intel Integrated Performance Primitives*, Intel Press, 2004.

[47] S. Müller and P. Massarani, "Transfer-function measurement with sweeps," *Journal of the Audio Engineering Society*, vol. 49, no. 6, pp. 443–471, 2001.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 179367, 11 pages
doi:10.1155/2010/179367

Research Article

PIC Detector for Piano Chords

Ana M. Barbancho, Lorenzo J. Tardón, and Isabel Barbancho

Departamento de Ingeniería de Comunicaciones, E.T.S. Ingeniería de Telecomunicación, Universidad de Málaga, Campus Universitario de Teatinos s/n, 29071 Málaga, Spain

Correspondence should be addressed to Isabel Barbancho, ibp@ic.uma.es

Received 22 February 2010; Revised 5 July 2010; Accepted 18 October 2010

Academic Editor: Xavier Serra

In this paper, a piano chords detector based on parallel interference cancellation (PIC) is presented. The proposed system makes use of the novel idea of modeling a segment of music as a third generation mobile communications signal, specifically, as a CDMA (Code Division Multiple Access) signal. The proposed model considers each piano note as a CDMA user in which the spreading code is replaced by a representative note pattern. The lack of orthogonality between the note patterns will make necessary to design a specific thresholding matrix to decide whether the PIC outputs correspond to the actual notes composing the chord or not. An additional stage that performs an octave test and a fifth test has been included that improves the error rate in the detection of these intervals that are specially difficult to detect. The proposed system attains very good results in both the detection of the notes that compose a chord and the estimation of the polyphony number.

1. Introduction

In this paper, we deal with a main stage of automatic music transcription systems [1]. We are refering to the detection of the notes that sound simultaneously in each of the temporal segments in which the musical piece can be divided. More precisely, we deal with the multiple fundamental frequency ($F0$) estimation problem in audio signals composed of piano chords. Therefore, the objective in this paper is to robustly determine the notes that sound simultaneously in each of the chords of a piano piece.

The approach employed in this paper is rather different from other proposals that can be found in the literature [1, 2]. In the paper by Goto [3], a multiple $F0$ estimation method based on a MAP approach to detect melody and bass lines is described. In the contribution by Klapuri [4, 5] a multiple $F0$ estimation method based on the iterative estimation of harmonic amplitudes and cancellation is presented. Kashino et al. [6, 7] propose a Bayesian approach to estimate notes and chords. Dixon [8] uses heuristics in the context of the Short Time Fourier Transform (STFT) to find peaks in the power spectrum to define musical notes; also tracking the detected peaks in consecutive audio segments is considered. In the paper by Tolonand and Karjalainen [9], a multipitch analysis model for audio and speech signals is proposed with some basis on the human auditory model. Vincent and Plumbley [10] propose an $F0$ extraction technique based on Bayesian harmonic models. Marolt [11, 12] uses a partial tracking technique based on a combination of an auditory model and adaptive oscillator networks followed by a time-delay neural network to perform automatic transcription of polyphonic piano music.

In this paper, we consider a different point of view. The audio signal to be analyzed will be considered to have certain similarities with the communications signal of a 3G mobile communications system. In this system, the communications signal is a code division multiple access (CDMA) signal [13]. This means that multiple signals from different users are transmitted simultaneously after a spreading process [14] that makes them approximately orthogonal signals. So, our model will consider each piano note as a CDMA user. We consider that the sinusoids with the frequencies of the partials of each note define a signal composed of approximately orthogonal components. In this signal, some of the sinusoidal components of the model, the effect of windowing, the time-variant nature of the music signal,

and other effects can be included in the concepts of noise and interference, that makes the different notes loose the property of orthogonality. So, each note will add interference (non orthogonal components) to other notes in a music signal in which several notes are simultaneously played. Then, the detection of the different notes played simultaneously can be considered as the problem of simultaneously removing the interference from the different notes and, then, deciding the notes played. The process is similar to the way in which a PIC receiver removes the interference from the multiple users to perform the symbol detection. In our context, the spreading codes will be the spectral patterns of the different notes. These patterns will include both the inherent characteristics of the piano and the style of the interpretation.

Turning back to the communications framework, it is clear that the most favorable and simplest case in CDMA systems is the one in which the spreading codes are orthogonal; that is, the cross-correlation between them is zero. In this case, it is known that the optimum detector is the conventional correlator. Then, the receiver can be easily implemented as a bank of filters adapted to the users' spreading codes [15]. Nevertheless, real CDMA systems do not fulfill the orthogonality condition; so the design of advanced detectors, like the PIC receiver, is required to cope with the interference due to the lack of orthogonality and to the multiuser access. In the context of musical signals, regarding the problem of detection of the notes that compose a musical chord, the orthogonality condition between the spectral patterns of the different notes cannot be achieved. This is due to the harmonic relations that exist between the notes of the equal-tempered musical scale typically used in Western music, specially between octaves and fifths (despite inharmonicity and stretched tuning [16]).

In order to perform the detection of the notes that sound in a certain segment or window of a musical audio signal, we have considered the CDMA detection technique called Parallel Interference Cancellation (PIC). We have selected PIC detection among other techniques [14, 15, 17] because it has been observed that PIC detection obtains very good performance in different CDMA system configurations [18] and it can be reasonably adapted to our problem. The PIC detector is aimed to simultaneously remove, for each user, the interference coming from the remaining users of the system. In the specific case of the music signal, regarding each piano note, the interference (parts or components of a note that are not orthogonal to other notes) caused by the rest of the notes should be simultaneously removed to allow the simultaneous detection of the different notes. A brief overview of the PIC detector for piano chords will be given in Section 2.

The paper is organized as follows. Section 2 will present a general view of the structure of the proposed PIC detector for piano chords. Section 3 will present the music signal model employed and the preprocessing techniques required, paying special attention to the similarities to CDMA signals. Section 3.1 will describe the process of estimation of the note patterns required to perform interference cancellation and detection and Section 3.2 will show the preprocessing tasks to be applied to the input signals before the interference cancellation process. Section 4 will describe in detail the structure of the interference cancellation stage of the parallel interference cancellation (PIC) detector adapted to the piano signal. Next, Section 5 will propose a method to finally decide the notes played using the outputs of the PIC. This section will cover not only the direct detection of notes but also specific tests to properly deal with their octaves and fifths. Section 6 will present some results and comparisons of the performance of the detection system. Finally, Section 7 will draw some conclusions.

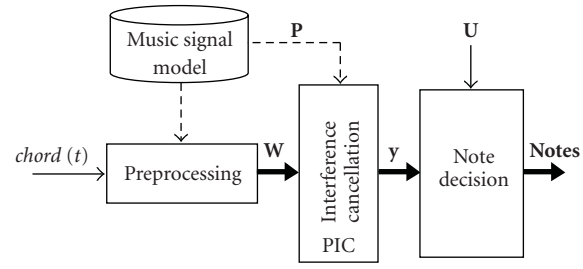


Figure 1: General structure of the PIC detector for piano chords.

2. Overview of the PIC Detector for Piano Chords

In this section, a general overview of the structure of a PIC Detector for piano chords is given. Figure 1 shows a general PIC structure in which the interference cancellation stage is the heart of the detector. The detector is defined upon three different stages.

The first stage (Preprocessing) obtains a representation of the chord ($chord(t)$) to be analyzed in the frequency domain so that its representation matches the signal model used in the system. Then, the preprocessed signal, **W**, passes through the parallel interference cancellation (PIC) block. This stage obtains an output for each of the notes of any piano ($L = 88$ notes for a standard piano). These values are related to the probability of having played each of the notes of the piano. To perform the parallel detection of interference, the note patterns (**P**) estimated from the musical signal model, taken as spreading codes, will be used. Finally, making use of the outputs of the PIC stage, **y**, it must be decided which are the notes that are actually present in the chord. This is the task of the final decision stage (Note Decision). This stage performs the decision using previously precomputed generic thresholds, **U**, together with a method of discrimination between actually played notes and octaves and fifths.

3. Music Signal Model

In this section, the music signal model considered to allow interference cancellation is presented. Also, marked similarities between the CDMA mobile communications signal and the audio signals are outlined. Recall that the music signals that will be handled by the proposed detector will be piano chords, that is, waveforms that contain the contribution of one or more notes that sound simultaneously. Consider a

piece (window) of the waveform of the music audio signal. This signal, say $chord(t)$, can be expressed, in general, as follows:

$$chord(t) = \sum_{n=1}^{M} A_n b_n p_n(t) + n(t), \quad (1)$$

where M, is the number of all the notes that can sound in the window (88 notes for a standard piano), A_n, represents the global amplitude of the nth note that may sound in the chord, $b_n \in \{1, 0\}$, indicates whether the note sounds (the note was played), $b_n = 1$, or not, $b_n = 0$, $p_n(t)$, stands for the representative waveform of the nth note in the chord, with normalized energy, and $n(t)$, represents additive white Gaussian noise (AWGN) with variance σ^2.

More details on this model will be given shortly, but before that, let's turn our sight to the mobile communications context. In such context, a certain window of a CDMA signal model can be expressed as follows [18]:

$$r(t) = \sum_{k=1}^{K} A_k b_k c_k(t) + n(t), \quad (2)$$

where K, is the number of simultaneous active users, A_k, is the amplitude of the kth user's signal, $b_k \in \pm 1$, is the bit transmitted by user k, $c_k(t)$, represents the spreading code assigned to user k with energy normalized to one, and $n(t)$, represents AWGN with variance σ^2.

A comparison of (1) and (2) reveals that they share the same formulation, but also some differences must be observed. In (2), the bits transmitted by user k, b_k, are represented by ± 1, while in (1) the values that b_n may take are 1 or 0. Moreover, at the sight of the two equations, the definition of $chord(t)$ takes into account all the possible notes that can be played, while $r(t)$, in (2), only includes the active users in the communications system (note that the number of possible user codes can be very high). Then, the problem of the detection of the notes played in a window of the available waveform, becomes the problem of deciding if b_n is 0 or 1 in (1), while the receiver of the communication system must detect the bits that have been transmitted by each active user, that is, to decide if b_k is 1 or -1. In spite of these differences, the similarity between (1) and (2) is enough to encourage us to consider the adaptation of advanced communication receivers to the detection of the notes in our musical context.

A main requirement of any CDMA detector is the following: the detector needs to know the spreading codes of the users, $c_k(t)$. In our context, according to (1), the partial waveforms of the notes, $p_n(t)$, are required, these will be called time patterns of the notes. But the same formulation is also valid in the frequency domain, then, the discrete power spectrum of $chord(t)$, can be expressed as follows:

$$W(k) = \sum_{n=1}^{M} A_n^2 b_n^2 P_n(k) + N(k), \quad (3)$$

where $P_n(k)$ is the kth bin of the power spectrum, $\mathbf{P}_n$, ($\mathbf{P}_n = [P_n(0), \ldots, P_n(k), \ldots, P_n(N-1)]^T$), of $p_n(t)$, and $N(k)$ represents the power spectrum of the additive noise $n(t)$.

It is clear that (3) is also similar to (2), in which the CDMA signal model is shown. If we consider a type of CDMA receiver adapted to our context, it will require to know the power spectrum, $P_n(k)$, of each of the notes that can sound in order to be able to perform the detection of the notes. These functions will be used to define the spectral patterns of the notes that will become the note patterns.

The audio signal model in the frequency domain will be used to design our system and the spectral patterns will be selected to represent the different notes just like spreading codes represent different users. The procedure to define the note patterns and the preprocessing stage required at the input of our PIC detector are described in the next subsections.

3.1. Determination of Note Patterns. In order to detect each note correctly, the detector needs to know the note patterns just like any CDMA detector needs to know the spreading codes of the users [19]. Also, these patterns should be as independent as possible of the piano and of the technique employed in the performance. Since the chord detection system will work in the frequency domain, spectral patterns of the notes will be used to play the role of the CDMA spreading codes in communication systems.

The representative spectral pattern of each note is obtained as the average power spectrum of 27 different waveforms of the possible performances in which each note can be played: three different playing techniques (Normal, Staccato and Pedal) in three different dynamics (Forte, Mezzo and Piano) and three different pianos. These samples are taken from the RWC data base [20], in which the audio signals are sampled at a frequency rate of 44.1 kHz and quantized with 16 bits. The length of the analysis windows, N, is also the number of bins of the power spectrum and it ranges between 2^{14} and 2^{17}, which results in analysis windows of duration between 371 ms and 2.97 s. These window lengths have been found adequate for a polyphonic music transcription system, showing a good compromise between time and frequency resolution [21]. The analysis windows are obtained applying a rectangular windowing function (simple truncation) to the signal waveform after the onset of the sound [22]. Note patterns are normalized to have unit energy so that they can be easily used in the interference cancellation stage (Section 4). With all this, each note pattern is a N-dimensional vector defined as:

$$\mathbf{P}_l = \frac{1}{Z_l} \sum_{i=1}^{N_p} \mathbf{P}_{l,i}, \quad (4)$$

where $\mathbf{P}_{l,i}$, is the vector that contains the N points of the power spectrum of the i-st performance of the l-st note of the piano, N_p, is the number of waveforms considered for each note (27 different performances per note), Z_l, is the normalization constant, defined as

$$Z_l = \sqrt{\left(\sum_{i=1}^{N_p} \mathbf{P}_{l,i}\right)^T \cdot \left(\sum_{i=1}^{N_p} \mathbf{P}_{l,i}\right)}. \quad (5)$$

In this way, general note patterns that take into account the positions of the partials and their relative power are obtained. These patterns can be used to detect the notes played in an analysis window regardless the piano employed and the interpretation technique. The set of patterns calculated for all piano notes will be denoted by **P:**

$$\mathbf{P} = \left[\mathbf{P}_1 \;\; \mathbf{P}_2 \;\; \cdots \;\; \mathbf{P}_M\right]. \tag{6}$$

This set of patterns will be used in the PIC detector as it will be described in Section 4.

The required signal preprocessing stage according to this audio signal model, is presented in the next subsection.

3.2. Preprocessing of Analysis Windows. Taking into account that the interference cancellation stage will perform in the frequency domain using the defined spectral note patterns, the detection system needs a stage to extract a representation of the signal that will be usable in the cancellation stage. This is the task of the preprocessing block in Figure 1.

The preprocessing stage obtains the discrete power spectrum of the windowed waveform under analysis (3) with length N, where N ranges between 2^{14} and 2^{17}, as in the process of determination of the note patterns (the windowing function used in this stage is the same that is used for the determination of the note patterns). The samples of the power spectrum are stored in the vector:

$$\mathbf{W} = [W(0), ..., W(N-1)]^T. \tag{7}$$

This vector constitutes the input to the parallel interference cancellation stage.

4. Parallel Interference Cancellation (PIC)

Once the note patterns are defined and stored in the pattern matrix **P**, and after the description of the preprocessing stage, the core of the detector, will be described.

A general description of the structure and behavior of PIC structures in communication systems together with comments on certain issues regarding to the cancellation policy, the receiver power of different users (notes in our context) and the number of cancellation stages can be found in [17]. Now, we will draw a description of the system specifically adapted to our context.

Figure 2 depicts the general structure of a linear multistage PIC detector, with m stages, for the detection of L-notes. Note that in our case $L = M$. This choice means that we will consider all the notes that can be played in a standard piano (88 notes from A0 to C8), unlike other authors that often do not consider the lowest and the highest octaves of the piano [4] (in [4] the range of notes detected is from E1 to C7). A general description of the behavior follows. Each note that sounds in the window under analysis ($chord(t)$), (**W** after preprocessing) introduces disturbance (interference) to the process of detection of each of the remaining $L - 1$ notes that may sound at the same time. Then, it should be possible to create replicas of the $L - 1$ notes detected to be simultaneously subtracted from the input signal (**W**) to remove their contribution (disturbance or interference) and to allow better performance of the note detection process at the next stage. This process is performed using the scheme in Figure 3. This figure will be described in detail later.

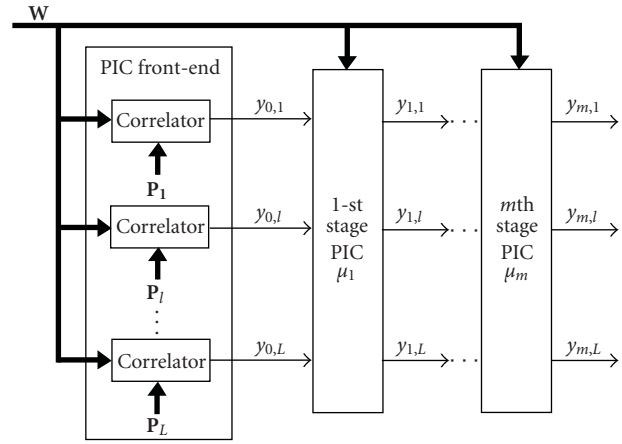


Figure 2: General structure of the PIC detector.

Note that if the initial detections are correct, then the replicas reconstructed could be perfect. This scheme would offer complete interference cancellation in one stage. On the other hand, if a note is detected, but it was not really sounding, a replica, created using the note patterns, subtracted from the input signal, adds additional disturbance (interference) to the process of detection of other notes. Also, any mismatch between a note pattern and the preprocessed waveform of that note may introduce interference into the detection process of other notes. This is a main reason why a more conservative procedure, in which interference is partially removed at successive interference cancellation stages, was proposed [23] and selected to deal with our problem (Figure 3). In this structure, as the stages progress, the detections should be more reliable and the cancellation process should be more accurate. Also, the unavoidable difference between the note patterns and the preprocessed note contributions to the chords discourages us from attempting to perform total interference cancellation.

Specifically, a multistage partial PIC detector structure has been chosen [17, 23, 24]. In this detector, the parameter μ_m, see Figures 2 and 3, represents the maximum amount of interference due to each note that will be canceled. In the context of digital communications systems, this strategy attains good performance with a small number of interference cancellation stages (between 3 and 7) when the weights of each stage, μ_m, are correctly chosen [18].

The interference cancellation structure, in our case, is analogous to the one presented in [18, 23]. Note that at the PIC front-end, an initial detection of the notes is performed using a bank of correlators. For each note l, the centered correlation between the preprocessed input signal, **W**, and the corresponding note pattern, $\mathbf{P}_l$, is calculated, $y_{0,l}$ (see Figure 2). The value obtained is used as input to the first cancellation stage (Figure 2).

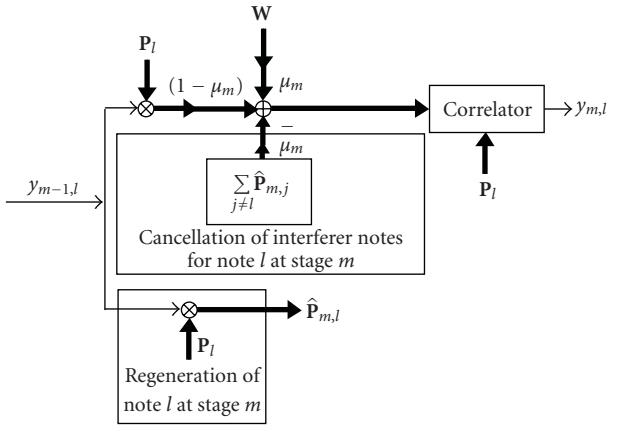


Figure 3: Stage m of the PIC detector for note l.

Now, the proper cancellation process starts. At each stage of the multistage PIC detector, for each note l, the process shown in Figure 3 is performed. In this figure, the following notation is employed.

(i) Thick lines represent vectors of length N, the length of the power spectrum considered.

(ii) Thin lines represent scalar values.

(iii) $\mathbf{P}_l$ is the pattern of the lth note of the piano, calculated using (4).

(iv) $l = 1, 2, \ldots, L$, where L is the number of piano notes considered (88 notes in our case).

(v) μ_m is the cancellation parameter for stage m. This parameter controls the amount of cancellation done at each stage. Usually, this parameter grows as the number of stage increases [25]. The reason for this choice is based on the expected improvement of the decision statistics obtained after each PIC stage as the signal goes through the interference cancellation system. Under this assumption, interference cancellation can be performed with lesser error in the successive stages.

(vi) $y_{m,l}$ is the decision statistic obtained for note l after the cancellation stage m.

(vii) *Correlator* calculates the centered correlation between the input signal and the note pattern $\mathbf{P}_l$.

(viii) $\hat{\mathbf{P}}_{m,l}$ represents the linear regeneration made at stage m of the possibly played note l. $\hat{\mathbf{P}}_{m,l}$ is given by

$$\hat{\mathbf{P}}_{m,l} = y_{m-1,l}\mathbf{P}_l. \tag{8}$$

As it can be observed in Figure 3, the output at each stage m for each note l is obtained by removing, from the preprocessed input $\mathbf{W}$, the regeneration of the remaining $(L-1)$ notes of the piano weighted by the cancellation parameter μ_m. The larger μ_m, the larger is the interference canceled.

Errors in the detections make the system add additional interference, instead of removing interference. The interference added in this case grows with the cancellation parameter. Therefore, the choice of cancellation weights is essential for the proper performance of the PIC. In Section 6, a comparison between different sets of weights and different number of stages shows the importance of the choice of these parameters.

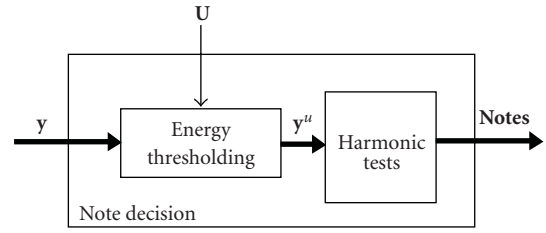


Figure 4: Structure of the Note Decision stage.

The output of the PIC for the detection of each note will be stored in the vector $\mathbf{y}$ (see Figures 1 and 2). This vector will contain the L decision statistics of the notes of the piano:

$$\mathbf{y} = \left[y_{m,1}, \ldots, y_{m,L}\right]^T. \tag{9}$$

This vector must be analyzed to decide which notes were played.

5. Played Note Decision

Making use of the PIC outputs, the system must decide which notes were played in the window under analysis. Ideally, the elements in $\mathbf{y}$ that correspond to the notes that were actually played, should be positive values and zero elsewhere. Unfortunately, this does not happen because of the windowing, the way in which the note patterns are defined, noise and because of the equal-tempered music scale, used in Western music. Note that assuming ideal harmonicity, the equal-tempered scale sets many nonorthogonal frequency relationships between different notes, being the most outstanding of them the octave and perfect fifth [21]. All these issues make appear significant values at the positions of the decision statistics obtained by the PIC for notes that were not actually played. The task of the Note Decision stage is to deal with this problem to make a decision on the notes played.

In Figure 4, the structure of the Note Decision stage is shown. This stage consists of two distinct blocks: Energy Thresholding and Harmonic Tests.

5.1. Energy Thresholding. The objective of this block is to identify the notes that definitely were not played. This initial decision is based on the comparison of the estimated energy of the contribution of each possible note to the (preprocessed) input signal $\mathbf{W}$, versus a threshold. In order to do this, all the decision statistics in $\mathbf{y}$ are compared with a threshold.

Now, the thresholds must be defined. In order to properly define them, we must first notice that before the normalization (see (5)), the note patterns of the different

notes do not have the same energy. The energy of the contribution of each note to the input signal will show the same behavior. So, the thresholds must take into account this feature. To this end, we decided to define thresholds for groups of notes clustered according to the mean energy of the samples available in our databases.

Let g denote the number of groups or clusters. We will define a matrix of thresholds, $\mathbf{U}$, for all the piano notes clustered in g groups. Note that these thresholds will be valid for all the notes regardless of the piano and the interpretation, just like the note patterns previously defined.

A detailed description of the process of creation of the groups of notes, the definition of the thresholds and how these thresholds are employed is now given:

Creation of the Clusters of Notes. First, we have to define the groups of notes that we will consider according to their expected mean energy. Recall that we refer to the selected representation of the notes in our system, not to the note waveforms. The mean energy of each note is calculated from the recorded samples of pianos 1 to 3 of the Musical Instrument Data Base RWC-MDB-1-2001-W01 [20]. We calculate the energy of each piano note played with different performance techniques and, then, the mean is obtained. Second, the notes are ordered according to their energy, in descendant order. The largest mean energy, M_e, is selected and the following energy interval is defined: $[0.66M_e, M_e]$ (the coefficient 0.66 has been experimentally obtained). The notes whose mean energy is in this interval compose the first group of notes. Then, these steps are recursively performed with the remaining notes until all the notes are grouped. After the completion of this process, $g = 6$ groups of notes are obtained.

Definition of Thresholds. We consider two types of threshold: one type of threshold for notes in the same group i (autothreshold, represented as u_{ii}) and the other one for the notes in the other groups j, where the group j has more energy than group i (it will be denoted crossthreshold and it will be represented as u_{ij}).

Autothresholds, u_{ii}, are calculated as follows: the notes with the largest and the lowest energy in the group i are selected (let i_E and i_e represent the indexes of these notes in the group i, resp.) and a composed signal formed summing the patterns of these notes ($\mathbf{P}_{i_E}$ and $\mathbf{P}_{i_e}$ resp.) weighted by the square root of their corresponding energy is obtained:

$$\mathbf{C}_{ii} = Z_{i_E}\mathbf{P}_{i_E} + Z_{i_e}\mathbf{P}_{i_e}, \tag{10}$$

where Z_x was defined in (5).

Then, this composed signal passes through the PIC detector (Figure 2). The vector obtained at the output of the PIC, $\mathbf{y}$, is normalized by the value of its largest element. Then, autothresholds are defined by the element in the normalized vector $\mathbf{y}$ that corresponds to the note with the lowest energy.

Crossthresholds, u_{ij}, are calculated in a similar way as autothresholds but different notes are selected as reference. Specifically, the note with the largest energy in the group j (j_E) and the note with the lowest energy in the group i(i_e) are selected. Then, the composed signal is defined as follows:

$$\mathbf{C}_{ij} = Z_{j_E}\mathbf{P}_{j_E} + Z_{i_e}\mathbf{P}_{i_e}. \tag{11}$$

This signal, $\mathbf{C}_{ij}$, passes through the PIC structure and, then, the threshold is defined as in the previous case.

Construction of the Matrix of Thresholds. All the thresholds defined are stored in a matrix with the following structure:

$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{22} & \cdots & u_{gg} \\ u_{21} & u_{22} & \cdots & u_{gg} \\ \vdots & & \ddots & \vdots \\ u_{g1} & u_{g2} & \cdots & u_{gg} \end{pmatrix}, \tag{12}$$

where each column represents all the thresholds found for a dominant group, j.

Usage of the Matrix of Thresholds. The group d, that contains the note with the largest value at the output of a PIC stage, $\mathbf{y}$, is selected. Then, the corresponding column of the matrix $\mathbf{U}$, $[u_{dd}, \ldots, u_{gd}]^T$, is used for thresholding.

Once the threshold column is selected, the elements in $\mathbf{y}$ under the corresponding thresholds are removed and the final decisions will be taken with the remaining elements.

The output of the energy thresholding block is denoted $\mathbf{y}^u$. This vector contains all the notes that were possibly sounding in the window under analysis. However, additional tests, that take into account harmonic relations among the notes, must be performed to avoid false positives.

5.2. Harmonic Tests. The last block of the note decision stage includes some harmonic tests to perform the final decision. One of the problems in polyphonic detection is the detection of the octave and perfect fifth since many errors occur due to either missing notes or, especially, to the appearance of false positives [26, 27]. This is due to the overlapping between harmonic partials of different sounds. Assuming ideal harmonicity, it is known that harmonic partials of two sounds coincide if and only if the fundamental frequencies of the two sounds are in rational number relations [28, 29]. When the harmonicity is not ideal, the overlapping continues since the partials of the notes may exhibit appreciable bandwidth. On the other hand, an important principle in Western music is that simple harmonic relationships are favored over dissonant ones in order to make the sounds blend better [21]. This is the case of octaves and fifths. These intervals are the ones whose harmonious relationships are the simplest (2 : 1 and 3 : 2) and these are also the two most frequent intervals in Western music [30].

The objective of the harmonic tests is to decide if the possibly played notes in $\mathbf{y}^u$ were actually played or if those are due to perfect octaves or perfect fifths. Finally, it is worth mentioning that this stage includes the estimation of the polyphony number in each chord.

In Figure 5, the general structure of the final stage is presented. The notation used in the figure is as follows.

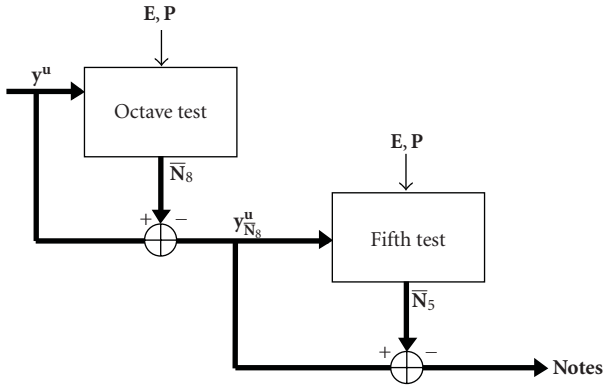


Figure 5: Structure of the harmonic tests.

(i) $\mathbf{y}^u$ is the vector that contains all the possibly played notes. It was obtained after the energy thresholding stage.

(ii) **E** is the vector that contains the mean energy of the 88 piano notes.

(iii) **P** is the note pattern matrix.

(iv) $\overline{\mathbf{N}}_8$ is the set of notes that do not pass the octave test.

(v) $\mathbf{y}^u_{\overline{\mathbf{N}}_8}$ is obtained removing from $\mathbf{y}^u$ the notes in $\overline{\mathbf{N}}_8$.

(vi) $\overline{\mathbf{N}}_5$ is the set of notes that do not pass the fifth test.

(vii) **Notes** is the final vector of notes detected.

As it can be seen in Figure 5, the decision process is as follows: first, all the possible notes with octave relations are considered and it is checked whether they are actually played notes. The notes that do not pass this test, $\overline{\mathbf{N}}_8$, are removed from $\mathbf{y}^u$ to define $\mathbf{y}_{\overline{N}_8}$. Then, all the possible notes with fifth relation in $\mathbf{y}_{\overline{N}_8}$, are considered and, then, it is checked if they are really played notes. Again, the notes that do not pass the test, $\overline{\mathbf{N}}_5$, are removed from $\mathbf{y}_{\overline{N}_8}$ to give a vector of notes detected (**Notes**).

5.2.1. Octave/Fifth Test. The octave and the fifth relation tests are similar, the only difference among them is the relation between the notes involved and the thresholds. Figure 6 shows the block diagram employed in the octave/fifth tests.

The notation used in Figure 6 is described now.

(i) $\mathbf{y}^u$ is the vector that contains all the possibly played notes.

(ii) $\mathbf{y}_x$ is the vector that contains a subset of notes from $\mathbf{y}^u$ or $\mathbf{y}_{\overline{N}_8}$ that fulfill the criteria of octave or fifth relation.

(iii) $(1/Z^u_x)\sum_{j=1, j\in\mathbf{y}_x}^{L} E_j\mathbf{P}_j$ is the signal composed with the patterns of the notes to check ($\mathbf{P}_j$) weighted by their corresponding energy (E_j), in order to properly cope with low- and high-energy notes, and normalized to unit energy using the normalization constant:

$$Z^u_x = \sqrt{\left(\sum_{\substack{j=1\\ j\in\mathbf{y}_x}}^{L} E_j\mathbf{P}_j\right)^T \cdot \left(\sum_{\substack{j=1\\ j\in\mathbf{y}_x}}^{L} E_j\mathbf{P}_j\right)} \tag{13}$$

(iv) $\mathbf{u}_{g,x}$ is the threshold vector for the octave/fifth-related notes.

(v) $\overline{\mathbf{N}}_x$ is the set of notes that do not pass the octave ($x = 8$)/fifth ($x = 5$) tests.

The operations performed in these tests are similar to those in the process of estimation of the thresholds $u_{g,x}$. The description of this process follows: a synthetic signal is composed with the patterns of the notes weighted by their corresponding energy. The synthetic signal is normalized to have unit energy. The composed signal passes through the PIC detector and the outputs are normalized by the maximum value of the outputs. Then, the output of the PIC, that correspond to the notes under test, are used as new thresholds for these notes. If a decision statistic of a note does not pass the new threshold, then the note will be removed from the set of possibly played notes since the value of the decision statistic found at the output of the PIC stage is considered to be due to some octave/fifth relation.

6. Results

The evaluation of the performance of the PIC detector for piano chords described in this paper and the comparison of the result versus a selected technique in [4] have been done using samples taken from different sources.

(i) Independent note samples: these samples correspond to pianos 1 to 3 of the Musical Instrument Data Base RWC-MDB-1-2001-W01 [20] and home-made recordings of two different pianos (Yamaha and Kawai).

(ii) Chord recordings: these samples are home made recordings of the two different pianos (Yamaha and Kawai).

The total number of samples available was over 4200. Note that the patterns are defined using a database which is different from the one used in the evaluation. The pianos used for the chord recordings are a Yamaha Clavinova CLP-130 and a Kawai CA91 played in a concert room.

The chords used to validate the system correspond, to the real chords frequently used in Western music. All the chords have been recorded in all the piano octaves and with different octave separations between the notes that constitute the chord. The recorded chords, as a function of the polyphony number, are as follows:

(i) chords of two notes: intervals of second, third, fourth, fifth and octaves as well as their extension with one, two, three and four octaves,

(ii) chords of three notes: perfect major and perfect minor chords with different order of notes,

(iii) chords of four notes: perfect major and perfect minor chords with duplication of their fundamental or their fifth, as well as, major 7th and minor 7th chords,

(iv) chords of five notes: perfect major and perfect minor chords with duplication of their fundamental and

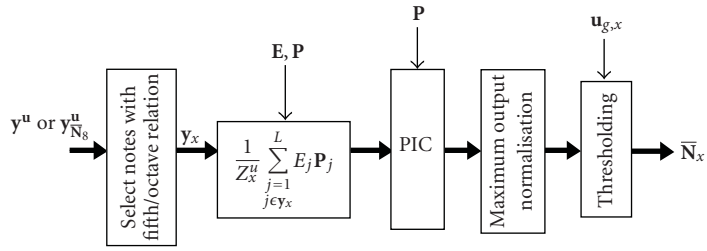


Figure 6: Block diagram of the octave/fifth test.

their fifth, as well as major 7th and minor 7th chords with duplication of their fundamental,

(v) chords of six notes: perfect major and perfect minor chords with duplication of their fundamental, their fifth and their third, as well as major 7th and minor 7th chords with duplication of their fundamental and their fifth. These chords have been always played with both hands and with a minimum separation of two octaves between the lowest note and the highest note. In most cases, this separation is four or five octaves, so the coincidences between partials of sounds with octave or fifth relation are smaller and the octave and fifth tests attain better performance.

The recorded chords satisfy the statistical profile discovered by Krumhansl in classical Western music [30], that is, octave relationships are the most frequently, followed by consonant musical intervals (perfect fifth, perfect fourth) and the smallest probability of occurrence is given to dissonant intervals (minor second, augmented fifth, etc.). Note that these are the types of chords actually used in Western music. In general, these chords are more difficult to resolve that the chords that are just composed with dissonant intervals [21].

The error measure employed is the note error rate (NER) metric. The NER is defined as the mean number of erroneously detected notes divided by the number of notes in the chords [21]:

$$\mathrm{NER} = \frac{\mathrm{SE} + \mathrm{DE} + \mathrm{IE}}{\mathrm{NN}}, \tag{14}$$

where Substitution errors (SE): happen when a note, that does not exist in the chord, is detected as played note, Deletion errors (DE): appear when the number of detected notes is smaller than the number of notes in a chord, Insertion errors (IE): appear when the number of detected notes is larger than the number of notes in a chord, NN: represents the number of notes in the chords.

It is worth mentioning that insertion errors (IE) never occurred in the proposed PIC detector in the tests done and the deletion errors only occur when the polyphony number is estimated.

Concerning the temporal resolution, windows with $N = 2^{14}$ samples were chosen. This choice gives a temporal resolution of about 371 ms and a spectral resolution of 2.69 Hz, which is the minimum resolution to distinguish the fundamental frequencies of the lowest notes of the piano.

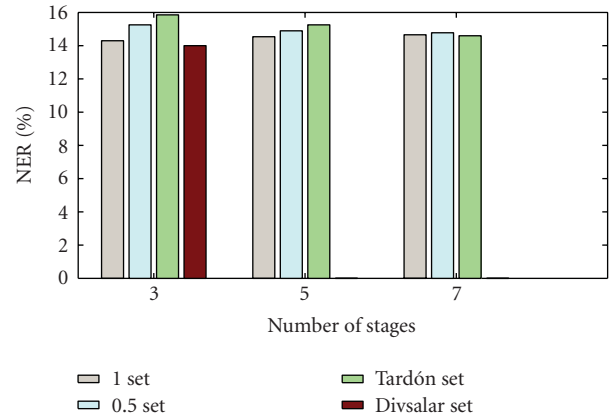


Figure 7: Comparison of note error rates for different sets of cancellation parameters and different number of parallel interference cancellation stages.

After several tests, and according to the results obtained for the CDMA signal in [18], a 3-stage PIC was chosen whose cancellation parameters are $\mu = [0.5, 0.7, 0.9]$ [23]. It has been observed that this choice provides a good balance between performance and complexity. A comparison of note error rates for PIC with 3, 5, or 7 stages and using 4 different sets of cancellation parameters are presented in Figure 7. The sets of cancellation parameters evaluated are as follows:

(i) "*1 set*": in this set, all the cancellation parameters are 1 (total interference cancellation is attempted at each stage) [31].

(ii) "*0.5 set*": in this set all the cancellation parameters are 0.5.

(iii) "*Tardón set*": in this set the cancellation parameters are defined as [25]:

$$\mu_k = \frac{1}{2}\frac{k}{K}, \tag{15}$$

where k is the stage and K the number of stages of the receiver.

(iv) "*Divsalar set*": in this set the cancellation parameters are $\mu = [0.5, 0.7, 0.9]$ [23].

Figure 7 shows that the cancellation parameters proposed by Divsalar attain the best NER. On the other hand, for "*1 set*" the NER increases with the number of stages, this is

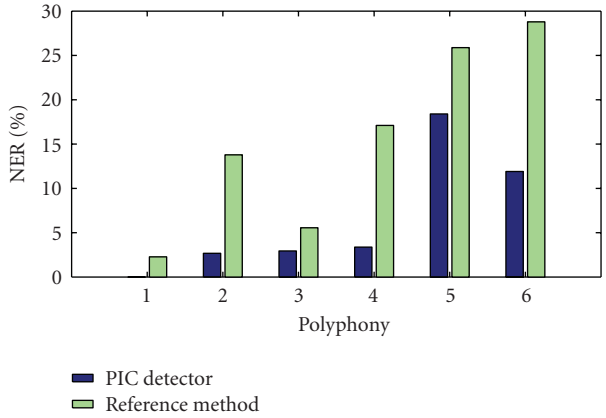


Figure 8: Comparison of note error rates for different polyphony numbers using the proposed PIC detector and the selected reference method proposed in [4]. Polyphony number known in both methods.

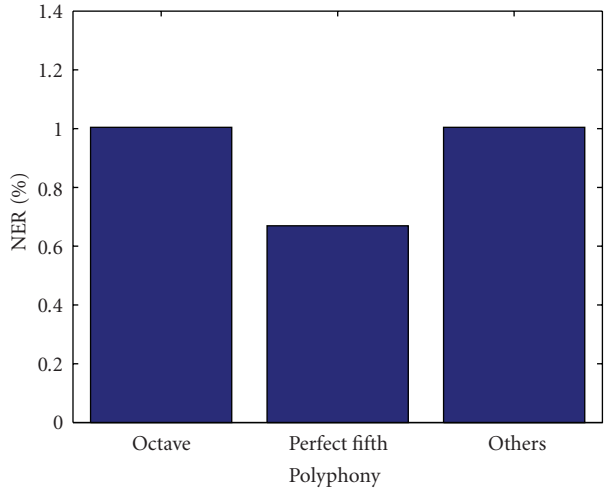


Figure 9: Note error rates for octaves, perfect fifths and other intervals using the proposed PIC detector when the polyphony number is 2.

due to the errors cancellation errors are accumulated because the cancellation in each stages is 100%. However, for "*0.5 set*" and "*Tardón set*" the NER decreases with the number of stages because the cancellation in each stage is small enough so that the cancellation errors do not negatively affect the detection performance of subsequent stages. Note that these sets require many interference cancellation stages (large computational burden) to attain the optimum performance which is attained with $K \rightarrow \infty$. However, the "*Divsalar set*", with interference cancellation stages, attains better performance than the other two sets of parameters with seven stages.

In Figure 8, a comparison of the NER for different polyphony numbers using the proposed PIC detector and the iterative estimation and cancellation reference method proposed in [4] is presented. In this case the polyphony number is known. Note that the method selected for comparison in [4] performs the detection of the notes in a successive way using a band wise F0 estimation for general purpose multiple F0 detection. However, our method performs the detection in a parallel way using specific note patterns. The dataset employed in the comparison was described at the beginning of this section.

It is worth mentioning that the errors are just substitution errors in both methods because the polyphony number is known. In this case, the output vector (**Notes**) is completed, if it is necessary, with the discarded notes in $\overline{\mathbf{N}}_8$ and $\overline{\mathbf{N}}_5$ for which the PIC output are larger. Recall that the proposed PIC detector never shows insertion errors and the deletion errors only occur when the polyphony number is estimated. As it can be observed in Figure 8, the NER increases with the polyphony number for both methods, however the proposed PIC detector gets better results and it can also deal with the low and high octaves of the piano. Note that the evaluation of the system in [4] is restricted to the range E1 to C7, because the F0s of the input dataset are restricted to that range. In this paper, we have evaluated, tuned and compared the systems in the range defined by all the piano notes. According to this choice, 12.5% of the notes are out of the range originally evaluated in [4].

There exists a gap in the performance between polyphony 4 and polyphony 5. This is due to the octave and fifth relations between the notes in these chords. In this case, the octave and fifth test sometimes fail when the chord includes several octaves and perfect fifths all together, because of the overlapping between the partials of more than three notes. On the other hand, the NER for a polyphony number of 6 is smaller than for polyphony number 5, the reason for this is the following: these chords have been always played with both hands and with a minimum of two octaves of separation between the lowest note and the highest note. In most cases, this separation is four or five octaves, so the coincidences between partials with octave or fifth relation are smaller and the octave and fifth test attain better performance.

If we also compare these results with the ones presented in [32], it should be taken into account that the evaluation of the system presented by Shi et al. [32] is made with sounds generated by mixing the sounds of different notes played solely and after a normalization of their amplitude to make the different notes of the same amplitude. However, the PIC detector proposed has been tested on recorded chords in which the different notes can be of different amplitudes and in which the chords are selected to be coherent and relevant from the musical point of view, as it has been presented at the beginning of this section.

Regarding the performance of the octave and fifth tests, Figure 9 represents the NER for octave, perfect fifth intervals and other intervals using the proposed PIC detector when the polyphony number is 2. In this figure, it can be observed that the NER for perfect fifth chords is smaller than the NER for octave intervals and other types of intervals.

Note that the fifth test performs better than the octave test because the overlap of the partials of the note patterns of notes with octave relation is larger than in the case of notes with fifth relation. Also, fifth test is performed after octave test. On the other hand, the NER for octaves is the same as

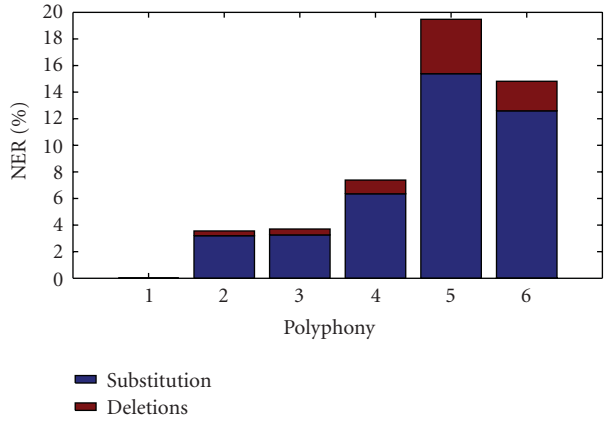


Figure 10: Note error rates for different polyphony numbers using the proposed PIC detector. Polyphony number estimated.

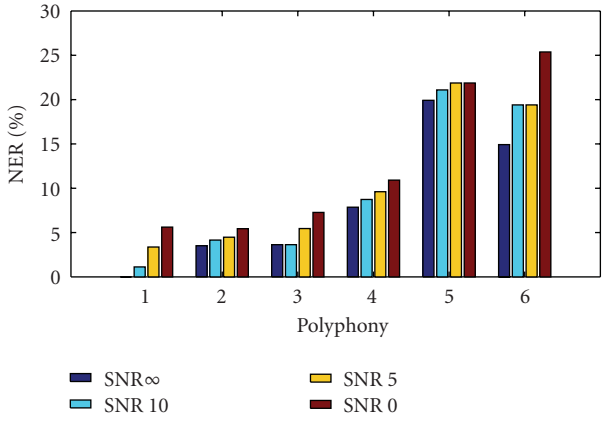


Figure 11: Note error rates in different levels of noise for different polyphony numbers using the proposed PIC detector. Polyphony number estimated.

for other types of intervals. These results show that the octave and fifth tests are efficient, making the errors to become almost independent of the type of interval that composed the chord under analysis.

Figure 10 shows the NER of the PIC detector when the polyphony number is estimated in the note decision block. As it can be observed, the NER is not significantly increased with respect to the case in which the polyphony number is known. In this figure, substitution and deletion errors are shown because, when the polyphony number is estimated, deletion errors can appear. It can be observed that the deletion errors are less than substitution errors. If we compare these results with the ones presented in Figure 8, it is clear that the increase of NER found when the polyphony number is estimated is mainly due to deletion errors.

If we compare the results in Figure 10 with the ones presented in [21] for the different polyphony estimation strategies, it can be observed that the proposed PIC detector attains better NER. Also, the difference in the performance between the cases in which the polyphony number is known and the cases in which it is estimated is smaller. This is an indication of the robustness of the proposed detection system both as note detector and as estimator of the degree of polyphony.

Figure 11 shows the note error rates in different levels of noise for different polyphony numbers using the proposed PIC detector when the polyphony number is estimated. No differences between substitution and deletion errors are shown because the percentage of deletion and substitution errors are the same as in Figure 10.

The noise variance has been selected so that the signal to noise ratio (SNR) is adjusted as in [21]. This figure shows that despite the NER increases with the noise, the proposed PIC system performs quite robustly in noisy cases. Again, the NER for a polyphony number of 6 is smaller than for polyphony number 5 because these chords have been always played with both hands, as previously described.

7. Conclusions

In this paper, a piano chords detector based on the idea of parallel interference cancellation has been presented. The proposed system makes use of the novel idea of modeling a segment of music as a third generation CDMA mobile communications signal. The model proposed considers each piano note as a CDMA user in which the spreading code is replaced by a representative note pattern defined in the frequency domain. This pattern is calculated by averaging the power spectral densities of different piano notes interpreted in various styles and with different pianos. This choice allows to attain good detection performance using these patterns regardless of the piano used to play the chord to be analyzed.

The structure of a multistage weighted PIC detector has been presented and it has been shown that the structure gets perfectly adapted to the purpose of the detection of the notes played in a chord. Since the spectral patterns of the notes are not orthogonal to each other, due to the harmonic relationships between the notes, and the different notes in a chord have different energies, a specific thresholding matrix has been designed for the task of deciding whether the PIC outputs correspond to real notes composing the chord. This matrix of thresholds is designed to be usable for any chord in any piano.

Finally, an additional stage that performs an octave test and a fifth test has been included. This stage eliminates false positives produced by the appearance of octave and fifth relations between the notes performed in the chord. It has been checked that these tests make the error rates in the detection of octaves and fifths to become similar to the ones found in the detection of any other type of interval.

The proposed system attains very good results in both the detection of the notes that compose a chord and the estimation of the polyphony number. Moreover, it has been observed that the detection performance is not noticeably affected by the estimation of the polyphony number with respect to the situations in which the polyphony number is known.

Acknowledgments

This work has been funded by the Ministerio de Educación y Ciencia of the Spanish Government under Project no. TSI2007-61181. The authors would like to thank Dr. Anssi Klapuri from Queen Mary University of London, UK, for offering them a reference method to compare the performance of the proposed system.

References

[1] A. P. Klapuri, "Automatic music transcription as we know it today," *Journal of New Music Research*, vol. 33, no. 3, pp. 269–282, 2004.

[2] S. W. Hainsworth, "Analysis of musical audio for polyphonic transcription," 1st year report, Department of Engineering, University of Cambridge, Cambridge, UK, 2001.

[3] M. Goto, "A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[4] A. P. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR '06)*, pp. 216–221, 2006.

[5] A. Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 255–266, 2008.

[6] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka, "Application of Bayesian probability network to music scene analysis," in *Proceedings of IJCAI Workshop on Computational Auditory Scene Analysis (CASA '95)*, pp. 52–59, 1995.

[7] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka, "Organization of hierarchical perceptual sounds: music scene analysis with autonomous processing modules and a quantitative information integration mechanism," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, vol. 1, pp. 158–164, Montreal, Canada, August 1995.

[8] S. Dixon, "Extraction of musical performance parameters from audio data," in *Proceedings of the 1st IEEE Pacific-Rim Conference on Multimedia*, pp. 42–45, 2000.

[9] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 708–716, 2000.

[10] E. Vincent and M. D. Plumbley, "Predominant-F0 estimation using Bayesian harmonic waveform models," in *Proceedings of the 1st Annual Music Information Retrieval Evaluation eXchange (MIREX '05)*, September 2005.

[11] M. Marolt, "Transcription of polyphonic piano music with neural networks," in *Proceedings of the 10th Mediterranean Electrotechnical Conference (MALECON '00)*, vol. 2, pp. 512–515, May 2000.

[12] M. Marolt, "A connectionist approach to automatic transcription of polyphonic piano music," *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 439–449, 2004.

[13] A. M. Barbancho, J. T. Entrambasaguas, and I. Barbancho, "CDMA systems physical function level simulator," in *Proceedings of the IASTED International Conference on Advances in Communications (AIC '01)*, pp. 61–66, 2001.

[14] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, USA, 3rd edition, 1995.

[15] S. Verdu, *Multiuser Detection*, Cambridge University Press, Cambridge, UK, 1998.

[16] T. D. Rossing, F. R. Moore, and P. A. Wheeler, *The Science of Sound*, Addison Wesley, San Francisco, Calif, USA, 3rd edition, 2002.

[17] D. Koulakiotis and A. H. Aghvami, "Data detection techniques for DS/CDMA mobile systems: a review," *IEEE Personal Communications*, vol. 7, no. 3, pp. 24–34, 2000.

[18] A. M. Barbancho, L. J. Tardón, and I. Barbancho, "Analytical performance analysis of the linear multistage partial PIC receiver for DS-CDMA systems," *IEEE Transactions on Communications*, vol. 53, no. 12, pp. 2006–2010, 2005.

[19] E. H. Dinan and B. Jabbari, "Spreading codes for direct sequence CDMA and wideband CDMA cellular networks," *IEEE Communications Magazine*, vol. 36, no. 9, pp. 48–54, 1998.

[20] M. Goto, "Development of the RWC music database," in *Proceedings of the 18th International Congress on Acoustics*, vol. 1, pp. 553–556, April 2004.

[21] A. P. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003.

[22] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

[23] D. Divsalar and M. K. Simon, "Improved parallel interference cancellation for CDMA," *IEEE Transactions on Communications*, vol. 46, no. 2, pp. 258–268, 1998.

[24] D. Guo, L. K. Rasmussen, S. Sun, T. J. Lim, and C. Cheah, "MMSE-based linear parallel interference cancellation in CDMA," in *Proceedings of the 5th IEEE International Symposium on Spread Spectrum Techniques and Applications*, vol. 3, pp. 917–921, September 1998.

[25] L. J. Tardón, E. Palacios, I. Barbancho, and A. M. Barbancho, "On the improved multistage partial parallel interference cancellation receiver for UMTS," in *Proceedings of the 60th IEEE Vehicular Technology Conference (VTC '04)*, vol. 4, pp. 2321–2325, September 2004.

[26] Y.-R. Chien and S.-K. Jeng, "An automatic transcription system with octave detection," in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP '02)*, vol. 2, pp. 1865–1868, Orlando, Fla, USA, May 2002.

[27] A. Schutz and D. Slock, "Periodic signal modeling for the octave problem in music transcription," in *Proceedings of the 16th International Conference on Digital Signal Processing (DSP '09)*, pp. 1–6, July 2009.

[28] G. Loy, *Musimathics, Volume 1: The Mathematical Foundations of Music*, The MIT Press, Cambridge, Mass, USA, 2006.

[29] J. Backus, *The Acoustical Foundations of Music*, W.W. Norton & Company, New York, NY, USA, 2nd edition, 1977.

[30] C. L. Krumhansl, *Cognitive Foundation of Musical Pitch*, Oxford University Press, New York, NY, USA, 1990.

[31] R. M. Buehrer and S. P. Nicoloso, "Comments on "partial parallel interference cancellation for CDMA"," *IEEE Transactions on Communications*, vol. 47, no. 5, pp. 658–661, 1999.

[32] L. Shi, J. Zhang, and G. Han, "Multiple fundamental frequency estimation based on harmonic structure model," in *Proceedings of the 2nd International Congress on Image and Signal Processing (CISP '09)*, pp. 1–4, Tianjin, China, October 2009.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 172961, 14 pages
doi:10.1155/2010/172961

Research Article

Query-by-Example Music Information Retrieval by Score-Informed Source Separation and Remixing Technologies

Katsutoshi Itoyama,[1] Masataka Goto,[2] Kazunori Komatani,[1] Tetsuya Ogata,[1] and Hiroshi G. Okuno[1]

[1] *Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Sakyo-Ku, Kyoto 606-8501, Japan*
[2] *Media Interaction Group, Information Technology Research Institute (ITRI), National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8568, Japan*

Correspondence should be addressed to Katsutoshi Itoyama, itoyama@kuis.kyoto-u.ac.jp

Received 1 March 2010; Revised 10 September 2010; Accepted 31 December 2010

Academic Editor: Augusto Sarti

We describe a novel query-by-example (QBE) approach in music information retrieval that allows a user to customize query examples by directly modifying the volume of different instrument parts. The underlying hypothesis of this approach is that the musical mood of retrieved results changes in relation to the volume balance of different instruments. On the basis of this hypothesis, we aim to clarify the relationship between the change in the volume balance of a query and the genre of the retrieved pieces, called *genre classification shift*. Such an understanding would allow us to instruct users in how to generate alternative queries without finding other appropriate pieces. Our QBE system first separates all instrument parts from the audio signal of a piece with the help of its musical score, and then it allows users remix these parts to change the acoustic features that represent the musical mood of the piece. Experimental results showed that the genre classification shift was actually caused by the volume change in the vocal, guitar, and drum parts.

1. Introduction

One of the most promising approaches in music information retrieval is query-by-example (QBE) retrieval [1–7], where a user can receive a list of musical pieces ranked by their similarity to a musical piece (example) that the user gives as a query. This approach is powerful and useful, but the user has to prepare or find examples of favorite pieces, and it is sometimes difficult to control or change the retrieved pieces after seeing them because another appropriate example should be found and given to get better results. For example, even if a user feels that vocal or drum sounds are too strong in the retrieved pieces, it is difficult to find another piece that has weaker vocal or drum sounds while maintaining the basic mood and timbre of the first piece. Since finding such music pieces is now a matter of trial and error, we need more direct and convenient methods for QBE. Here we assume that QBE retrieval system takes audio inputs and treat low-level acoustic features (e.g., Mel-frequency cepstral coefficients, spectral gradient, etc.).

We solve this inefficiency by allowing a user to create new query examples for QBE by remixing existing musical pieces, that is, changing the volume balance of the instruments. To obtain the desired retrieved results, the user can easily give alternative queries by changing the volume balance from the piece's original balance. For example, the above problem can be solved by customizing a query example so that the volume of the vocal or drum sounds is decreased. To remix an existing musical piece, we use an original sound source separation method that decomposes the audio signal of a musical piece into different instrument parts on the basis of its musical score. To measure the similarity between the remixed query and each piece in a database, we use the Earth Movers Distance (EMD) between their Gaussian Mixture

Models (GMMs). The GMM for each piece is obtained by modeling the distribution of the original acoustic features, which consist of intensity and timbre.

The underlying hypothesis is that changing the volume balance of different instrument parts in a query grows diversity of the retrieved pieces. To confirm this hypothesis, we focus on the musical genre since musical diversity and musical genre have a certain level of relationship. A music database that consists of various genre pieces is suitable for the purpose. We define the term *genre classification shift* as the change of musical genres in the retrieved pieces. We target genres that are mostly defined by organization and volume balance of musical instruments, such as classical music, jazz, and rock. We exclude genres that are defined by specific rhythm patterns and singing style, e.g., waltz and hip hop. Note that this does not mean that the genre of the query piece itself can be changed. Based on this hypothesis, our research focuses on clarifying the relationship between the volume change of different instrument parts and the shift in the musical genre of retrieved pieces in order to instruct a user in how to easily generate alternative queries. To clarify this relationship, we conducted three different experiments. The first experiment examined how much change in the volume of a single instrument part is needed to cause a genre classification shift using our QBE retrieval system. The second experiment examined how the volume change of two instrument parts (a two-instrument combination for volume change) cooperatively affects the shift in genre classification. This relationship is explored by examining the genre distribution of the retrieved pieces. These experimental results show that the desired genre classification shift in the QBE results was easily achieved by simply changing the volume balance of different instruments in the query. The third experiment examined how the source separation performance affects the shift. The retrieved pieces using sounds separated by our method are compared with those using original sounds before mixing down in producing musical pieces. The experimental result showed that the separation performance for predictable feature shifts depends on an instrument part.

2. Query-by-Example Retrieval by Remixed Musical Audio Signals

In this section, we describe our QBE retrieval system for retrieving musical pieces based on the similarity of mood between musical pieces.

2.1. Genre Classification Shift. Our original term "*genre classification shift*" means a change in the musical genre of pieces based on auditory features, which is caused by changing the volume balance of musical instruments. For example, by boosting the vocal and reducing the guitar and drums of a popular song, auditory features are extracted from the modified song are similar to the features of a jazz song. The instrumentation and volume balance of musical instruments affects the musical mood. The musical genre does not have direct relation to the musical mood but genre classification shift in our QBE approach suggests that remixing query examples grow the diversity of retrieved results. As shown in Figure 1, by automatically separating the original recording (audio signal) of a piece into musical instrument parts, a user can change the volume balance of these parts to cause a genre classification shift.

2.2. Acoustic Feature Extraction. Acoustic features that represent the musical mood are designed as shown in Table 1 upon existing studies of mood extraction [8]. These features extracted from the power spectrogram, $X(t, f)$, for each frame (100 frames per second). The spectrogram is calculated by short-time Fourier transform of the monauralized input audio signal, where t and f are the frame and frequency indices, respectively.

2.2.1. Acoustic Intensity Features. Overall intensity for each frame, $S_1(t)$, and intensity of each subband, $S_2(i,t)$, are defined as

$$S_1(t) = \sum_{f=1}^{F_N} X(t,f), \qquad S_2(i,t) = \sum_{f=F_L(i)}^{F_H(i)} X(t,f), \tag{1}$$

where F_N is the number of frequency bins of the power spectrogram and $F_L(i)$ and $F_H(i)$ are the indices of lower and upper bounds for the ith subband, respectively. The intensity of each subband helps to represent acoustic brightness. We use octave filter banks that divide the power spectrogram into n octave subbands:

$$\left[1, \frac{F_N}{2^{n-1}}\right), \left[\frac{F_N}{2^{n-1}}, \frac{F_N}{2^{n-2}}\right), \ldots, \left[\frac{F_N}{2}, F_N\right], \tag{2}$$

where n is the number of subbands, which is set to 7 in our experiments. These filter banks cannot be constructed because they have ideal frequency response; we implemented these by division and sum of the power spectrogram.

2.2.2. Acoustic Timbre Features. Acoustic timbre features consist of spectral shape features and spectral contrast features, which are known to be effective in detecting musical moods [8, 9]. The spectral shape features are represented by spectral centroid $S_3(t)$, spectral width $S_4(t)$, spectral rolloff $S_5(t)$, and spectral flux $S_6(t)$:

$$\begin{aligned} S_3(t) &= \frac{\sum_{f=1}^{F_N} X(t,f) f}{S_1(t)}, \\ S_4(t) &= \frac{\sum_{f=1}^{F_N} X(t,f) \left(f - S_3(t)\right)^2}{S_1(t)}, \\ \sum_{f=1}^{S_5(t)} X(t,f) &= 0.95 S_1(t), \\ S_6(t) &= \sum_{f=1}^{F_N} \left(\log X(t,f) - \log X(t-1,f)\right)^2. \end{aligned} \tag{3}$$

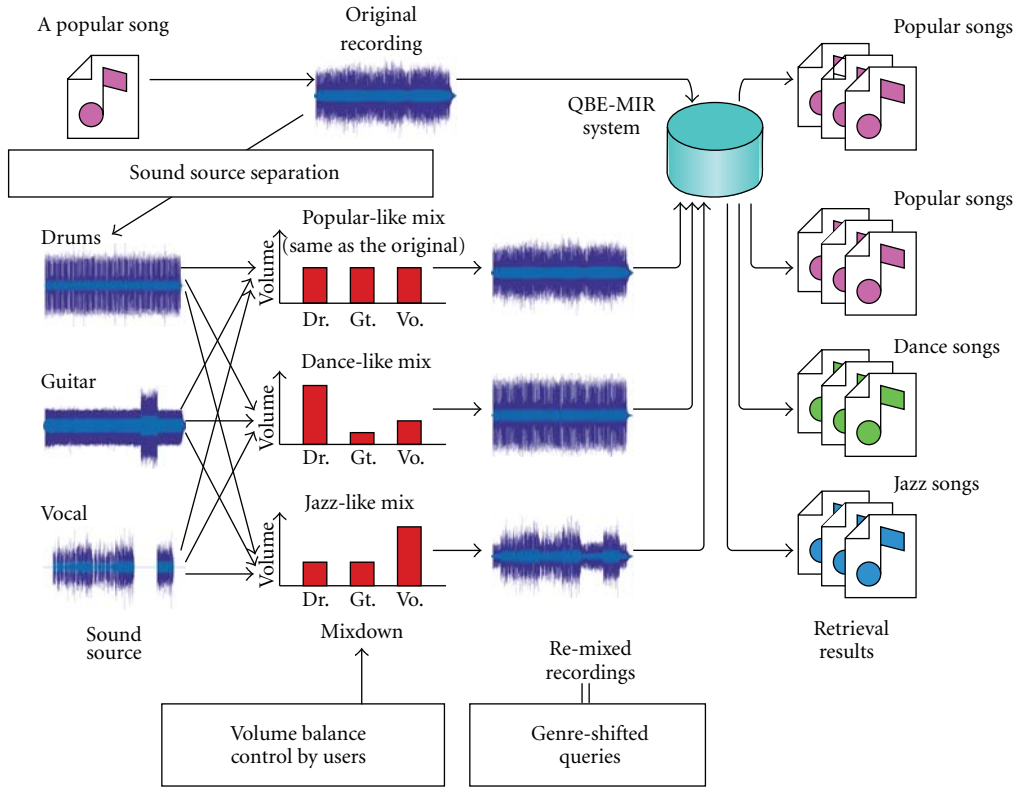


Figure 1: Overview of QBE retrieval system based on genre classification shift. Controlling the volume balance causes a genre classification shift of a query song, and our system returns songs that are similar to the genre-shifted query.

Table 1: Acoustic features representing musical mood.

Acoustic intensity features		
Dim.	Symbol	Description
1	$S_1(t)$	Overall intensity
2–8	$S_2(i,t)$	Intensity of each subband*
Acoustic timbre features		
Dim.	Symbol	Description
9	$S_3(t)$	Spectral centroid
10	$S_4(t)$	Spectral width
11	$S_5(t)$	Spectral rolloff
12	$S_6(t)$	Spectral flux
13–19	$S_7(i,t)$	Spectral peak of each subband*
20–26	$S_8(i,t)$	Spectral valley of each subband*
27–33	$S_9(i,t)$	Spectral contrast of each subband*

* 7-band octave filter bank.

The spectral contrast features are obtained as follows. Let a vector,

$$(X(i,t,1), X(i,t,2), \ldots, X(i,t,F_N(i))), \quad (4)$$

be the power spectrogram in the tth frame and ith subband. By sorting these elements in descending order, we obtain another vector,

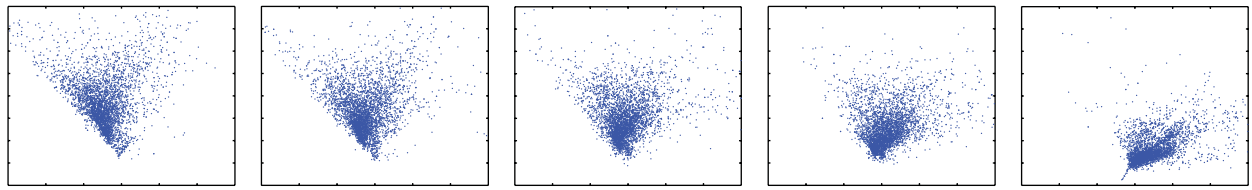
$$(X'(i,t,1), X'(i,t,2), \ldots, X'(i,t,F_N(i))), \quad (5)$$

where

$$X'(i,t,1) > X'(i,t,2) > \cdots > X'(i,t,F_N(i)) \quad (6)$$

as shown in Figure 3 and $F_N(i)$ is the number of the ith subband frequency bins:

$$F_N(i) = F_H(i) - F_L(i). \quad (7)$$



(a) $-\infty$ dB (b) -5 dB (c) ± 0 dB (d) $+5$ dB (e) $+\infty$ dB

Figure 2: Distributions of the first and second principal components of extracted features from the no. 1 piece of the RWC Music Database: Popular Music. Five figures show the shift of feature distribution by changing the volume of the drum part. The shift of feature distribution causes the genre classification shift.

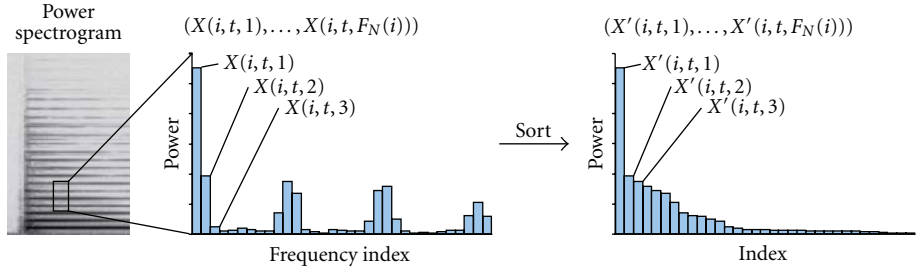


Figure 3: Sorted vector of power spectrogram.

Here, the spectral contrast features are represented by spectral peak $S_7(i,t)$, spectral valley $S_8(i,t)$, and spectral contrast $S_9(i,t)$:

$$S_7(i,t) = \log\left(\frac{\sum_{f=1}^{\beta F_N(i)} X'(i,t,f)}{\beta F_N(i)}\right),$$
$$S_8(i,t) = \log\left(\frac{\sum_{f=(1-\beta)F_N(i)}^{F_N(i)} X'(i,t,f)}{\beta F_N(i)}\right), \quad (8)$$
$$S_9(i,t) = S_7(i,t) - S_8(i,t),$$

where β is a parameter for extracting stable peak and valley values, which is set to 0.2 in our experiments.

2.3. Similarity Calculation. Our QBE retrieval system needs to calculate the similarity between musical pieces, that is, a query example and each piece in a database, on the basis of the overall mood of the piece.

To model the mood of each piece, we use a Gaussian Mixture Model (GMM) that approximates the distribution of acoustic features. We set the number of mixtures to 8 empirically, although a previous study [8] used a GMM with 16 mixtures since we used smaller database than that study for experimental evaluation. Although the dimension of the obtained acoustic features was 33, it was reduced to 9 by using the principal component analysis where the cumulative percentage of eigenvalues was 0.95.

To measure the similarity among feature distributions, we utilized Earth Movers Distance (EMD) [10]. The EMD is based on the minimal cost needed to transform one distribution into another one.

3. Sound Source Separation Using Integrated Tone Model

As mentioned in Section 1, musical audio signals should be separated into instrument parts beforehand to boost and reduce the volume of those parts. Although a number of sound source separation methods [11–14] have been studied, most of them still focus on dealing with music performed on either pitched instruments that have harmonic sounds or drums that have inharmonic sounds. For example, most separation methods for harmonic sounds [11–14] cannot separate inharmonic sounds, while most separation methods for inharmonic sounds, such as drums [15], cannot separate harmonic ones. Sound source separation methods based on the stochastic properties of audio signals, for example, independent component analysis and sparse coding [16–18], treat particular kind of audio signals which are recorded with a microphone array or have small number of simultaneously voiced musical notes. However, these methods cannot separate complex audio signals such as commercial CD recordings. We describe our sound source separation method which can separate complex audio signals with both harmonic and inharmonic sounds in this section.

The input and output of our method are described as follows:

input power spectrogram of a musical piece and its musical score (standard MIDI file); standard MIDI files for famous songs are often available thanks to Karaoke applications; we assume the spectrogram and the score have already been aligned (synchronized) by using another method;

output decomposed spectrograms that correspond to each instrument.

To separate the power spectrogram, we approximate the power spectrogram which is purely additive. By playing back each track of the SMF on a MIDI sound module, we prepared a sampled sound for each note. We call this a template sound and used it as prior information (and initial values) in the separation. The musical audio signal corresponding to the decomposed power spectrogram is obtained by using the inverse short-time Fourier transform with the phase of the input spectrogram.

In this section, we first define the problem of separating sound sources and the integrated tone model. This model is based on a previous study [19], and we improved implementation of the inharmonic models. We then derive an iterative algorithm that consists of two steps: sound source separation and model parameter estimation.

3.1. Integrated Tone Model of Harmonic and Inharmonic Models. Separating the sound source means decomposing the input power spectrogram, $X(t,f)$, into a power spectrogram that corresponds to each musical note, where t and f are the time and the frequency, respectively. We assume that $X(t,f)$ includes K musical instruments and the kth instrument performs L_k musical notes.

We use an integrated tone model, $J_{kl}(t,f)$, to represent the power spectrogram of the lth musical note performed by the kth musical instrument ((k,l)th *note*). This tone model is defined as the sum of harmonic-structure tone models, $H_{kl}(t,f)$, and inharmonic-structure tone models, $I_{kl}(t,f)$, multiplied by the whole amplitude of the model, $w_{kl}^{(J)}$:

$$J_{kl}(t,f) = w_{kl}^{(J)}\left(w_{kl}^{(H)}H_{kl}(t,f) + w_{kl}^{(I)}I_{kl}(t,f)\right), \quad (9)$$

where $w_{kl}^{(J)}$ and $(w_{kl}^{(H)}, w_{kl}^{(I)})$ satisfy the following constraints:

$$\sum_{k,l} w_{kl}^{(J)} = \iint X(t,f)\,dt\,df, \qquad \forall k,l : w_{kl}^{(H)} + w_{kl}^{(I)} = 1. \quad (10)$$

The harmonic tone model, $H_{kl}(t,f)$, is defined as a constrained two-dimensional Gaussian Mixture Model (GMM), which is a product of two one-dimensional GMMs, $\sum u_{klm}^{(H)}E_{klm}^{(H)}(t)$ and $\sum v_{kln}^{(H)}F_{kln}^{(H)}(f)$. This model is designed by referring to the HTC source model [20]. Analogously, the inharmonic tone model, $I_{kl}(t,f)$, is defined as a constrained two-dimensional GMM that is a product of two one-dimensional GMMs, $\sum u_{klm}^{(I)}E_{klm}^{(I)}(t)$ and $\sum v_{kln}^{(I)}F_{kln}^{(I)}(f)$. The temporal structures of these tone models, $E_{klm}^{(H)}(t)$ and $E_{klm}^{(I)}(t)$, are defined as an identical mathematical formula, but the frequency structures, $F_{kln}^{(H)}(f)$ and $F_{kln}^{(I)}(f)$, are defined as different forms. In the previous study [19], the inharmonic models are implemented in a nonparametric way. We changed the inharmonic model by implementing in a parametric way. This change improves generalization of the integrated tone model, for example, timbre modeling and extension to a bayesian estimation.

The definitions of these models are as follows:

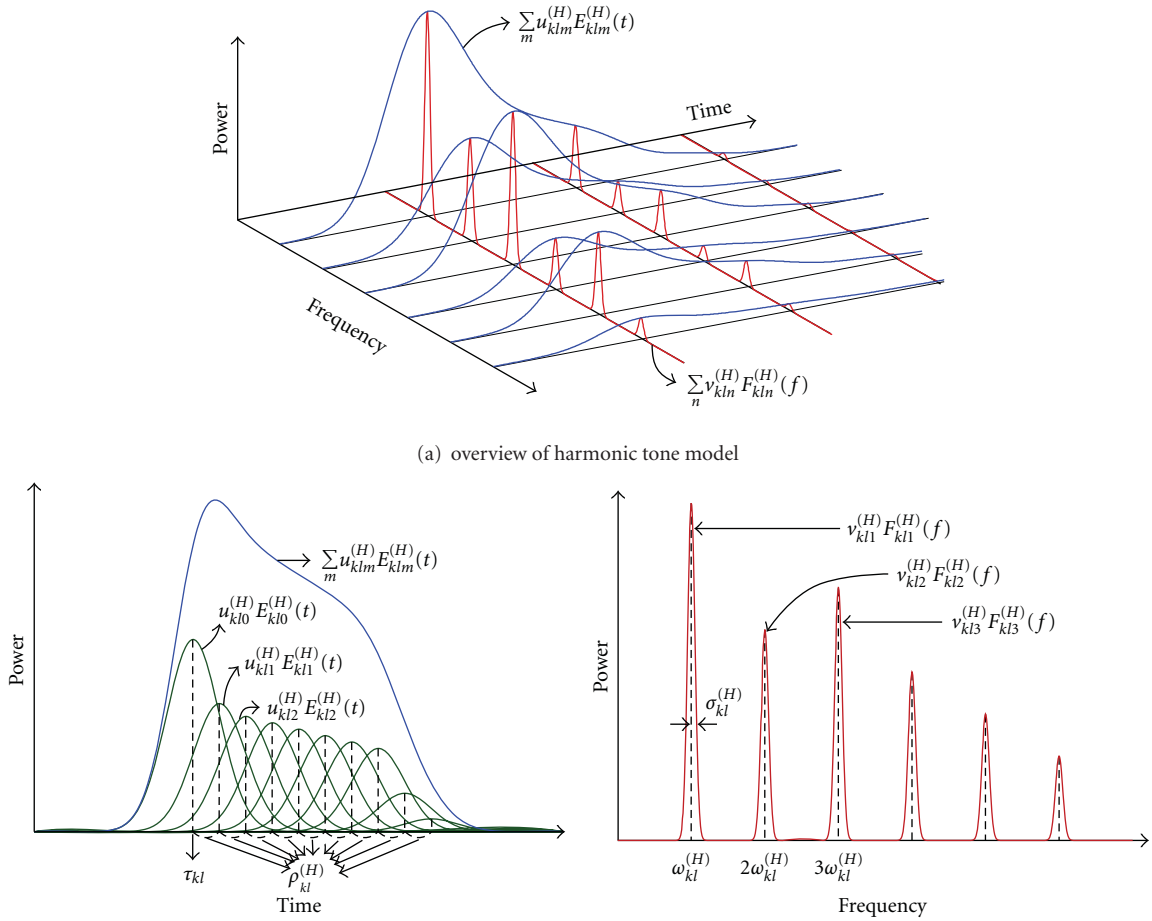
$$\begin{aligned}
H_{kl}(t,f) &= \sum_{m=0}^{M_H-1}\sum_{n=1}^{N_H} u_{klm}^{(H)}E_{klm}^{(H)}(t)v_{kln}^{(H)}F_{kln}^{(H)}(f),\\
I_{kl}(t,f) &= \sum_{m=0}^{M_I-1}\sum_{n=1}^{N_I} u_{klm}^{(I)}E_{klm}^{(I)}(t)v_{kln}^{(I)}F_{kln}^{(I)}(f),\\
E_{klm}^{(H)}(t) &= \frac{1}{\sqrt{2\pi}\rho_{kl}^{(H)}}\exp\left(-\frac{\left(t-\tau_{klm}^{(H)}\right)^2}{2\left(\rho_{kl}^{(H)}\right)^2}\right),\\
F_{kln}^{(H)}(f) &= \frac{1}{\sqrt{2\pi}\sigma_{kl}^{(H)}}\exp\left(-\frac{\left(f-\omega_{kln}^{(H)}\right)^2}{2\left(\sigma_{kl}^{(H)}\right)^2}\right),\\
E_{klm}^{(I)}(t) &= \frac{1}{\sqrt{2\pi}\rho_{kl}^{(I)}}\exp\left(-\frac{\left(t-\tau_{klm}^{(I)}\right)^2}{2\left(\rho_{kl}^{(I)}\right)^2}\right),\\
F_{kln}^{(I)}(f) &= \frac{1}{\sqrt{2\pi}(f+\kappa)\log\beta}\exp\left(-\frac{(\mathcal{F}(f)-n)^2}{2}\right),\\
\tau_{klm}^{(H)} &= \tau_{kl} + m\rho_{kl}^{(H)},\\
\omega_{kln}^{(H)} &= n\omega_{kl}^{(H)},\\
\tau_{klm}^{(I)} &= \tau_{kl} + m\rho_{kl}^{(I)},\\
\mathcal{F}(f) &= \frac{\log((f/\kappa)+1)}{\log\beta}.
\end{aligned} \quad (11)$$

All parameters of $J_{kl}(t,f)$ are listed in Table 2. Here, M_H and N_H are the numbers of Gaussian kernels that represent temporal and frequency structures of the harmonic tone model, respectively, and M_I and N_I are the numbers of Gaussians that represent those of the inharmonic tone model. β and κ are coefficients that determine the arrangement of Gaussian kernels for the frequency structure of the inharmonic model. If $1/(\log\beta)$ and κ are set to 1127 and 700, $\mathcal{F}(f)$ is equivalent to the mel scale of f Hz. Moreover $u_{klm}^{(H)}$, $v_{kln}^{(\mathrm{H})}$, $u_{klm}^{(I)}$, and $v_{kln}^{(I)}$ satisfy the following conditions:

$$\begin{aligned}
\forall k,l :\ &\sum_m u_{klm}^{(H)} = 1,\\
\forall k,l :\ &\sum_n v_{kln}^{(H)} = 1,\\
\forall k,l :\ &\sum_m u_{klm}^{(I)} = 1,\\
\forall k,l :\ &\sum_n v_{kln}^{(I)} = 1.
\end{aligned} \quad (12)$$

As shown in Figure 5, function $F_{kln}^{(I)}(f)$ is derived by changing the variables of the following probability density function:

$$\mathcal{N}(g;n,1) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(g-n)^2}{2}\right), \quad (13)$$

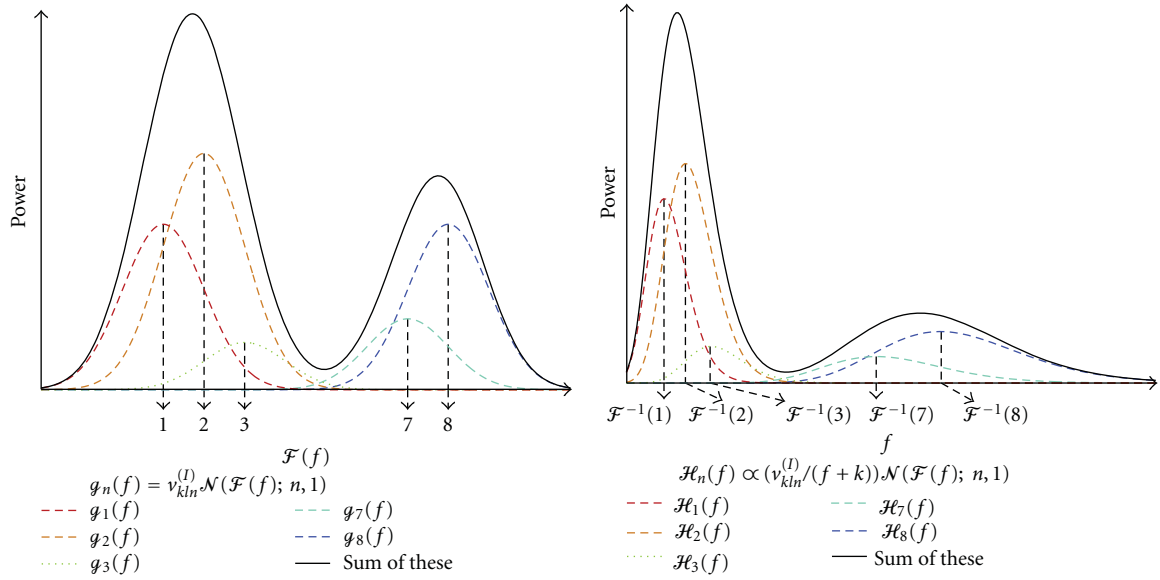


(a) overview of harmonic tone model

(b) temporal structure of harmonic tone model

(c) frequency structure of harmonic tone model

FIGURE 4: Overall, temporal, and frequency structures of the harmonic tone model. This model consists of a two-dimensional Gaussian Mixture Model, and it is factorized into a pair of one-dimensional GMMs.



(a) Equally-spaced Gaussian kernels along the log-scale frequency, $\mathcal{F}(f)$.

(b) Gaussian kernels obtained by changing the random variables of the kernels in (a).

FIGURE 5: Frequency structure of inharmonic tone model.

Table 2: Parameters of integrated tone model.

Symbol	Description
$w_{kl}^{(J)}$	Overall amplitude
$w_{kl}^{(H)}$, $w_{kl}^{(I)}$	Relative amplitude of harmonic and inharmonic tone models
$u_{klm}^{(H)}$	Amplitude coefficient of temporal power envelope for harmonic tone model
$v_{kln}^{(H)}$	Relative amplitude of the nth harmonic component
$u_{klm}^{(I)}$	Amplitude coefficient of temporal power envelope for inharmonic tone model
$v_{kln}^{(I)}$	Relative amplitude of the nth inharmonic component
τ_{kl}	Onset time
$\rho_{kl}^{(H)}$	Diffusion of temporal power envelope for harmonic tone model
$\rho_{kl}^{(I)}$	Diffusion of temporal power envelope for inharmonic tone model
$\omega_{kl}^{(H)}$	F0 of harmonic tone model
$\sigma_{kl}^{(H)}$	Diffusion of harmonic components along frequency axis
β, κ	Coefficients that determine the arrangement of the frequency structure of inharmonic model

from $g = \mathcal{F}(f)$ to f, that is,

$$\begin{aligned} F_{kln}^{(I)}(f) &= \frac{dg}{df}\mathcal{N}(\mathcal{F}(f); n, 1) \\ &= \frac{1}{(f+\kappa)\log\beta}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(\mathcal{F}(f)-n)^2}{2}\right). \end{aligned} \tag{14}$$

3.2. Iterative Separation Algorithm. The goal of this separation is to decompose $X(t, f)$ into each (k, l)th note by multiplying a spectrogram distribution function, $\Delta^{(J)}(k, l; t, f)$, that satisfies

$$\begin{aligned} &\forall k, l, t, f : 0 \le \Delta^{(J)}(k, l; t, f) \le 1, \\ &\forall t, f : \sum_{k,l}\Delta^{(J)}(k, l; t, f) = 1. \end{aligned} \tag{15}$$

With $\Delta^{(J)}(k, l; t, f)$, the separated power spectrogram, $X_{kl}^{(J)}(t, f)$, is obtained as

$$X_{kl}^{(J)}(t, f) = \Delta^{(J)}(k, l; t, f)X(t, f). \tag{16}$$

Then, let $\Delta^{(H)}(m, n; k, l, t, f)$ and $\Delta^{(I)}(m, n; k, l, t, f)$ be spectrogram distribution functions that decompose $X_{kl}^{(J)}(t, f)$ into each Gaussian distribution of the harmonic and inharmonic models, respectively. These functions satisfy

$$\begin{aligned} &\forall k, l, m, n, t, f : 0 \le \Delta^{(H)}(m, n; k, l, t, f) \le 1, \\ &\forall k, l, m, n, t, f : 0 \le \Delta^{(I)}(m, n; k, l, t, f) \le 1, \end{aligned} \tag{17}$$

$$\begin{aligned} \forall k, l, t, f : 0 \le &\sum_{m,n}\Delta^{(H)}(m, n; k, l, t, f) \\ &+ \sum_{m,n}\Delta^{(I)}(m, n; k, l, t, f) = 1. \end{aligned} \tag{18}$$

With these functions, the separated power spectrograms, $X_{klmn}^{(H)}(t, f)$ and $X_{klmn}^{(I)}(t, f)$, are obtained as

$$\begin{aligned} X_{klmn}^{(H)}(t, f) &= \Delta^{(H)}(m, n; k, l, t, f)X_{kl}^{(J)}(t, f), \\ X_{klmn}^{(I)}(t, f) &= \Delta^{(I)}(m, n; k, l, t, f)X_{kl}^{(J)}(t, f). \end{aligned} \tag{19}$$

To evaluate the effectiveness of this separation, we use an objective function defined as the Kullback-Leibler (KL) divergence from $X_{klmn}^{(H)}(t, f)$ and $X_{klmn}^{(I)}(t, f)$ to each Gaussian kernel of the harmonic and inharmonic models:

$$\begin{aligned} Q^{(\Delta)} = \sum_{k,l}\Bigg(&\sum_{m,n}\iint X_{klmn}^{(H)}(t, f) \\ &\times \log\frac{X_{klmn}^{(H)}(t, f)}{u_{klm}^{(H)}v_{kln}^{(H)}E_{klm}^{(H)}(t)F_{kln}^{(H)}(f)}\,dt\,df \\ &+ \sum_{m,n}\iint X_{klmn}^{(I)}(t, f) \\ &\times \log\frac{X_{klmn}^{(I)}(t, f)}{u_{klm}^{(I)}v_{kln}^{(I)}E_{klm}^{(I)}(t)F_{kln}^{(I)}(f)}\,dt\,df\Bigg). \end{aligned} \tag{20}$$

The spectrogram distribution functions are calculated by minimizing $Q^{(\Delta)}$ for the functions. Since the functions satisfy the constraint given by (18), we use the method of Lagrange multiplier. Since $Q^{(\Delta)}$ is a convex function for the spectrogram distribution functions, we first solve the simulteneous equations, that is, derivatives of the sum of $Q^{(\Delta)}$ and Lagrange multipliers for condition (18) are equal to zero, and then obtain the spectrogram distribution functions,

$$\begin{aligned} \Delta^{(H)}(m, n; k, l, t, f) &= \frac{E_{klm}^{(H)}(t)F_{kln}^{(H)}(f)}{\sum_{k,l}J_{kl}(t, f)}, \\ \Delta^{(I)}(m, n; k, l, t, f) &= \frac{E_{klm}^{(I)}(t)F_{kln}^{(I)}(f)}{\sum_{k,l}J_{kl}(t, f)}, \end{aligned} \tag{21}$$

and decomposed spectrograms, that is, separated sounds, on the basis of the parameters of the tone models.

Once the input spectrogram is decomposed, the likeliest model parameters are calculated using a statistical estimation. We use auxiliary objective functions for each (k,l)th note, $Q_{k,l}^{(Y)}$, to estimate robust parameters with power spectrogram of the template sounds, $Y_{kl}(t,f)$. The (k,l)th auxiliary objective function is defined as the KL divergence from $Y_{klmn}^{(H)}(t,f)$ and $Y_{klmn}^{(I)}(t,f)$ to each Gaussian kernel of the harmonic and inharmonic models:

$$Q_{k,l}^{(Y)} = \sum_{m,n} \iint Y_{klmn}^{(H)}(t,f) \log \frac{Y_{klmn}^{(H)}(t,f)}{u_{klm}^{(H)} v_{kln}^{(H)} E_{klm}^{(H)}(t) F_{kln}^{(H)}(f)} dt\, df + \sum_{m,n} \iint Y_{klmn}^{(I)}(t,f) \log \frac{Y_{klmn}^{(I)}(t,f)}{u_{klm}^{(I)} v_{kln}^{(I)} E_{klm}^{(I)}(t) F_{kln}^{(I)}(f)} dt\, df, \tag{22}$$

where

$$Y_{klmn}^{(H)}(t,f) = \Delta^{(H)}(m,n;k,l,t,f)\, Y_{kl}(t,f), \qquad Y_{klmn}^{(I)}(t,f) = \Delta^{(I)}(m,n;k,l,t,f)\, Y_{kl}(t,f). \tag{23}$$

Then, let Q be a modified objective function that is defined as the weighted sum of $Q^{(\Delta)}$ and $Q_{k,l}^{(Y)}$ with weight parameter α:

$$Q = \alpha Q^{(\Delta)} + (1-\alpha) \sum_{k,l} Q_{k,l}^{(Y)}. \tag{24}$$

We can prevent the overtraining of the models by gradually increasing α from 0 (i.e., the estimated model should first be close to the template spectrogram) through the iteration of the separation and adaptation (model estimation). The parameter update equations are derived by minimizing Q. We experimentally set α to 0.0, 0.25, 0.5, 0.75, and 1.0 in sequence and 50 iterations are sufficient for parameter convergence with each alpha value. Note that this modification of the objective function has no direct effect on the calculation of the distribution functions since the modification never changes the relationship between the model and the distribution function in the objective function. For all α values, the optimal distribution functions are calculated from only the models written in (21). Since the model parameters are changed by the modification, the distribution functions are also changed indirectly. The parameter update equations are described in the appendix.

We obtain an iterative algorithm that consists of two steps: calculating the distribution function while the model parameters are fixed and updating the parameters under the distribution function. This iterative algorithm is equivalent to the Expectation-Maximization (EM) algorithm on the basis of the maximum *a posteriori* estimation. This fact ensures the local convergence of the model parameter estimation.

4. Experimental Evaluation

We conducted two experiments to explore the relationship between instrument volume balances and genres. Given the query musical piece in which the volume balance is changed, the genres of the retrieved musical pieces are investigated. Furthermore, we conducted an experiment to explore the influence of the source separation performance on this relationship, by comparing the retrieved musical pieces using clean audio signals before mixing down (*original*) and separated signals (*separated*).

Table 3: Number of musical pieces for each genre.

Genre	Number of pieces
Popular	6
Rock	6
Dance	15
Jazz	9
Classical	14

Ten musical pieces were excerpted for the query from the *RWC Music Database: Popular Music* (RWC-MDB-P-2001 no. 1–10) [21]. The audio signals of these musical pieces were separated into each musical instrument part using the standard MIDI files, which are provided as the AIST annotation [22]. The evaluation database consisted of 50 other musical pieces excerpted from the *RWC Music Database: Musical Genre* (RWC-MDB-G-2001). This excerpted database includes musical pieces in the following genres: popular, rock, dance, jazz, and classical. The number of pieces are listed in Table 3.

In the experiments, we reduced or boosted the volumes of three instrument parts—vocal, guitar, and drums. To shift the genre of the retrieved musical piece by changing the volume of these parts, the part of an instrument should have sufficient duration. For example, the volume of an instrument that is performed for 5 seconds in a 5-minute musical piece may not affect the genre of the piece. Thus, the above three instrument parts were chosen because they satisfy the following two constraints:

(1) played in all 10 musical pieces for the query,

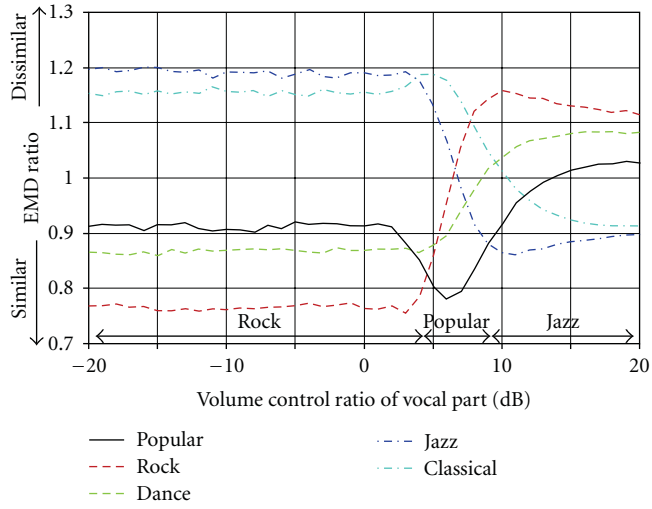
(2) played for more than 60% of the duration of each piece.

At http://winnie.kuis.kyoto-u.ac.jp/~itoyama/qbe/, sound examples of remixed signals and retrieved results are available.

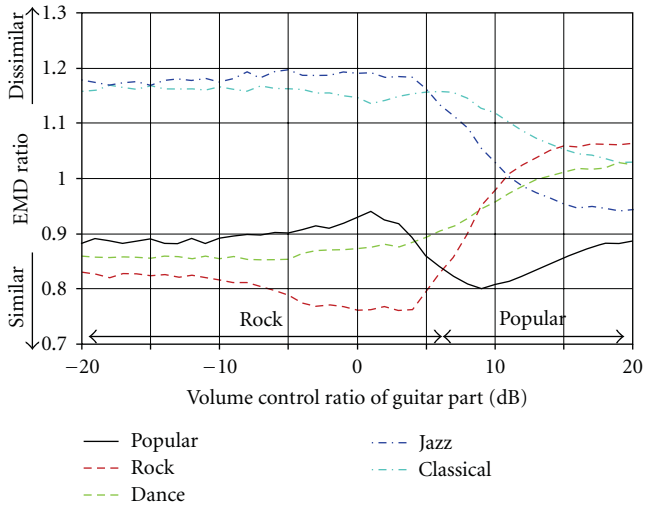
4.1. Volume Change of Single Instrument. The EMDs were calculated between the acoustic feature distributions of each query song and each piece in the database as described in Section 2.3, while reducing or boosting the volume of these musical instrument parts between 20 and +20 dB. Figure 6 shows the results of changing the volume of a single instrument part. The vertical axis is the relative ratio of the EMD averaged over the 10 pieces, which is defined as

$$\text{EMD ratio} = \frac{\text{average EMD of each genre}}{\text{average EMD of all genres}}. \tag{25}$$

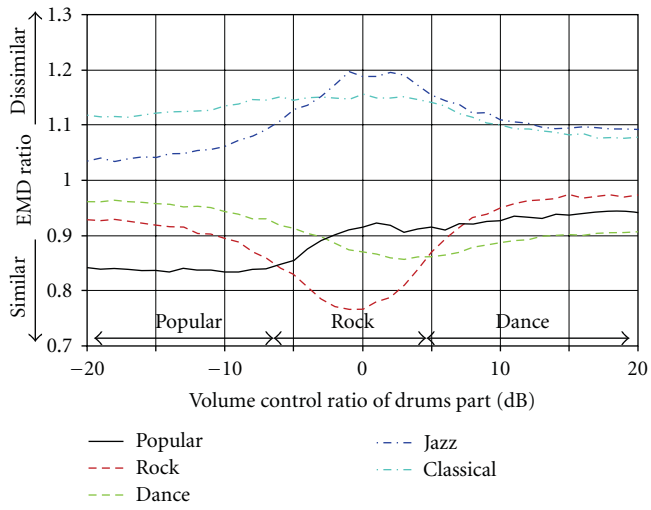
The results in Figure 6 clearly show that the genre classification shift occurred by changing the volume of any



(a) genre classification shift caused by changing the volume of vocal. Genre with the highest similarity changed from rock to popular and to jazz



(b) genre classification shift caused by changing the volume of guitar. Genre with the highest similarity changed from rock to popular

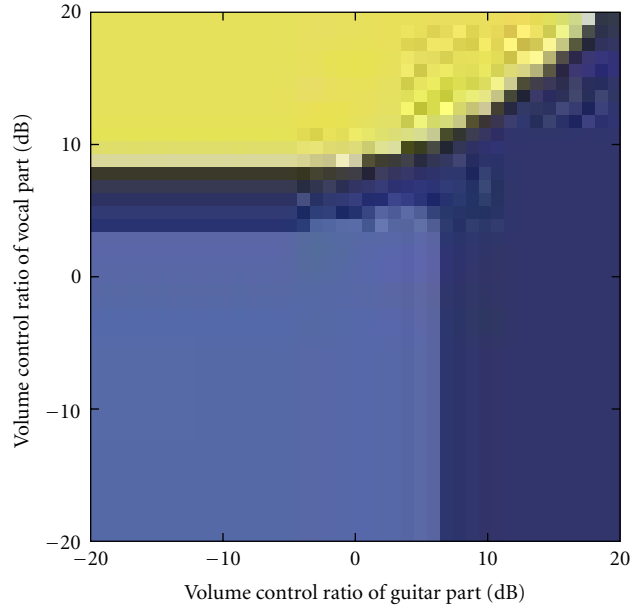


(c) genre classification shift caused by changing the volume of drums. Genre with the highest similarity changed from popular to rock and to dance

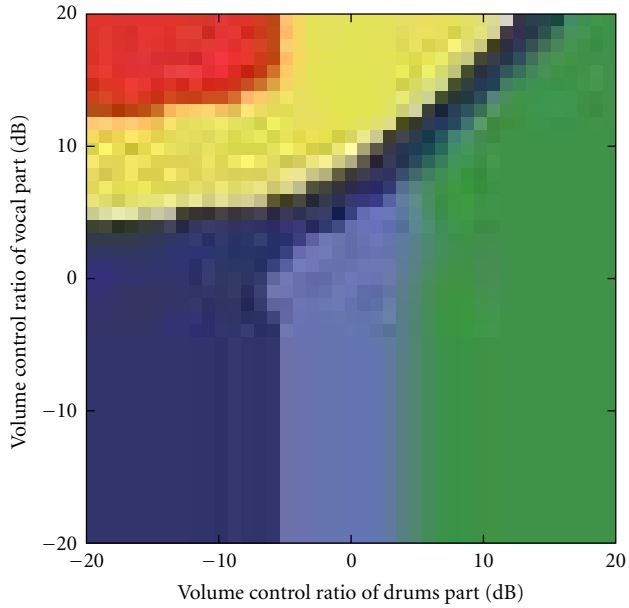
Figure 6: Ratio of average EMD per genre to average EMD of all genres while reducing or boosting the volume of single instrument part. Here, (a), (b), and (c) are for the vocal, guitar, and drums, respectively. Note that a smaller ratio of the EMD plotted in the lower area of the graph indicates higher similarity. (a) Genre classification shift caused by changing the volume of vocal. Genre with the highest similarity changed from rock to popular and to jazz. (b) Genre classification shift caused by changing the volume of guitar. Genre with the highest similarity changed from rock to popular. (c) Genre classification shift caused by changing the volume of drums. Genre with the highest similarity changed from popular to rock and to dance.

instrument part. Note that the genre of the retrieved pieces at 0 dB (giving the original queries without any changes) is the same for all three Figures 6(a), 6(b), and 6(c). Although we used 10 popular songs excerpted from the *RWC Music Database: Popular Music* for the queries, they are considered to be rock music as the genre with the highest similarity at 0 dB because those songs actually have the true rock flavor with strong guitar and drum sounds.

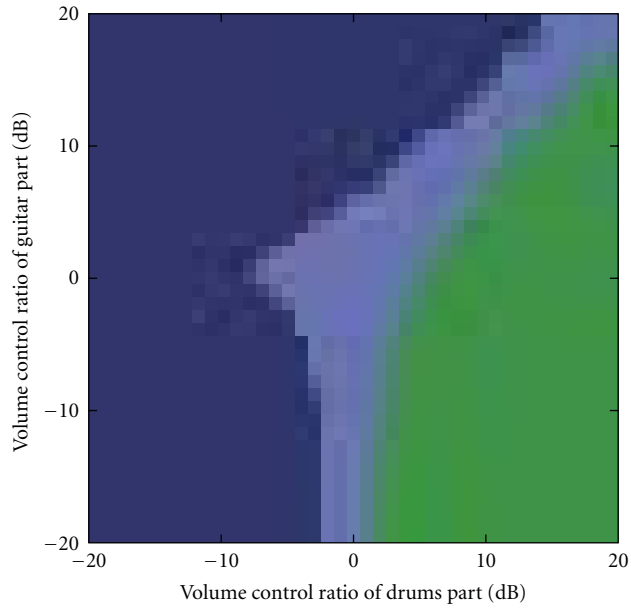
By increasing the volume of the vocal from −20 dB, the genre with the highest similarity shifted from rock (−20 to 4 dB) to popular (5 to 9 dB) and to jazz (10 to 20 dB) as shown in Figure 6(a). By changing the volume of the guitar, the genre shifted from rock (−20 to 7 dB) to popular (8 to 20 dB) as shown in Figure 6(b). Although it was commonly observed that the genre shifted from rock to popular in both cases of vocal and guitar, the genre shifted to jazz only in the case of vocal. These results indicate that the vocal and guitar would have differentimportance in jazz music. By changing the volume of the drums, genres shifted from popular (−20 to −7 dB) to rock (−6 to 4 dB) and to dance (5 to 20 dB)



(a) genre classification shift caused by changing the volume of vocal and guitar



(b) genre classification shift caused by changing the volume of vocal and drums



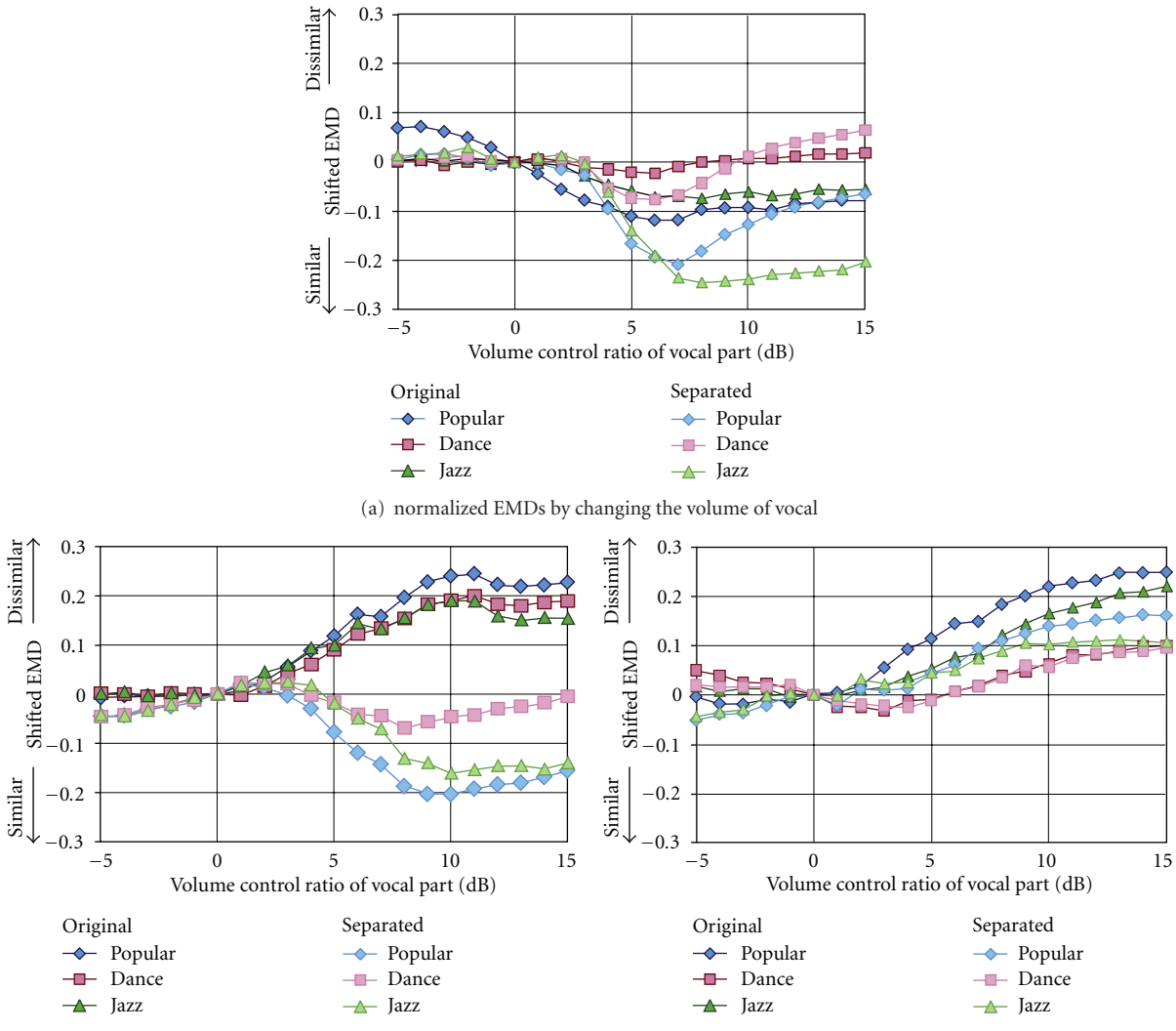
(c) genre classification shift caused by changing the volume of guitar and drums

Figure 7: Genres that have the smallest EMD (the highest similarity) while reducing or boosting the volume of two instrument parts. (a), (b), and (c) are the cases of the vocal-guitar, vocal-drums, and guitar-drums, respectively. (a) Genre classification shift caused by changing the volume of vocal and guitar. (b) Genre classification shift caused by changing the volume of vocal and drums. (c) Genre classification shift caused by changing the volume of guitar and drums.

as shown in Figure 6(c). These results indicate a reasonable relationship between the instrument volume balance and the genre classification shift, and this relationship is consistent with typical impressions of musical genres.

4.2. Volume Change of Two Instruments (Pair). The EMDs were calculated in the same way as the previous experiment. Figure 7 shows the results of simultaneously changing the volume of two instrument parts (instrument pairs). If one of the parts is not changed (at 0 dB), the results are the same as those in Figure 6.

Although the basic tendency in the genre classification shifts is similar to the single instrument experiment, classical music, which does not appear as the genre with the highest



(a) normalized EMDs by changing the volume of vocal

(b) normalized EMDs by changing the volume of guitar

(c) normalized EMDs by changing the volume of drums

Figure 8: Normalized EMDs that are shifted to 0 when the volume control ratio is 0 dB for each genre while reducing or boosting the volume. (a), (b), and (c) graphs are obtained by changing the volume of the vocal, guitar, and drum parts, respectively. Note that a smaller EMD plotted in the lower area of each graph indicates higher similarity than the one without volume controlling. (a) Normalized EMDs by changing the volume of vocal. (b) Normalized EMDs by changing the volume of guitar. (c) Normalized EMDs by changing the volume of drums.

similarity in Figure 6, appears in Figure 7(b) when the vocal part is boosted and the drum part is reduced. The similarity of rock music decreased when we *separately* boosted either the guitar or the drums, but it is interesting that rock music can keep the highest similarity if both the guitar and drums are boosted *together* as shown in Figure 7(c). This result closely matched with the typical impression of rock music, and it suggests promising possibilities for this technique as a tool for customizing the query for QBE retrieval.

4.3. Comparison between Original and Separated Sounds. The EMDs were calculated while reducing or boosting the volume of the musical instrument parts between −5 and +15 dB. Figure 8 shows the normalized EMDs that are shifted to 0 when the volume control ratio is 0 dB. Since all query songs are popular music, EMDs between query songs and popular pieces in the evaluation database tend to be smaller than the pieces of other genres. In this experiment, EMDs were normalized because we focused on the shifts in the acoustic features.

By changing the volume of the drums, the EMDs plotted in Figure 8(c) have similar curves in both of the *original* and *separated* conditions. On the other hand, by changing the volume of the guitar, the EMDs plotted in Figure 8(b) showed that a curve of the original condition is different from a curve of the separation condition. This result indicates that the shifts of features in those conditions were different. Average source separation performance of the guitar part was −1.77 dB, which was a lower value than those of vocal and drum parts. Noises included in the separated sounds

of the guitar part induced this difference. By changing the volume of the vocal, the plotted EMDs of popular and dance pieces have similar curves, but the EMDs of jazz pieces have different curves, although the average source separation performance of the vocal part is the highest among these three instrument parts. This result indicates that the separation performance for predictable feature shifts depends on the instrument part.

5. Discussions

The aim of this paper is achieving a QBE approach which can retrieve diverse musical pieces by boosting or reducing the volume balance of the instruments. To confirm the performance of the QBE approach, evaluation using a music database which has wide variations is necessary. A music database that consists of various genre pieces is suitable for the purpose. We defined the term *genre classification shift* as the change of musical genres in the retrieved pieces since we focus on the diversity of the retrieved pieces not on musical genre change of the query example.

Although we conducted objective experiments to evaluate the effectiveness of our QBE approach, several questions remain as open questions.

(1) More evidences of our QBE approach by subjective experiments are needed whether the QBE retrieval system can help users search better results.
(2) In our experiments, we used only popular musical pieces as query examples. Remixing query examples except popular pieces can shift genres of retrieved results.

For source separation, we use the MIDI representation of a musical signal. Mixed and separated musical signals contain variable features: timbre difference from musical instruments' individuality, characteristic performances of instrument players such as vibrato, and environments such as room reverberation and sound effects. These features can be controlled implicitly by changing the volume of musical instruments and therefore QBE systems can retrieve various musical pieces. Since MIDI representations do not contain these features, diversity of retrieved musical pieces will decrease and users cannot evaluate the mood difference of the pieces if we use only musical signals which are synthesized from MIDI representations.

In the experiments, we used precisely synchronized SMFs at most 50 milliseconds of onset timing error. In general, synchronization between CD recordings and their MIDI representations is not enough for separation. Previous studies on audio-to-MIDI synchronization methods [23, 24] can help this problem. We experimentally confirmed that onset timing error under 200 milliseconds does not decrease source separation performance. Another problem is that the proposed separation method needs a complete musical score with melody and accompaniment instruments. A study of source separation method with a MIDI representation of specified instrument part [25] will help solving the accompaniment problem.

In this paper, we aimed to analyze and decompose a mixture of harmonic and inharmonic sounds by appending the inharmonic model to the harmonic model. To achieve this, a requirement must be satisfied: one-to-one basis-source mapping based on structured and parameterized source model. The HTC source model [20], on which our integrated model is based, satisfies the requirement. Adaptive harmonic spectral decomposition [26] has modeled a harmonic structure in a different way. They are suitable for multiple-pitch analysis and applied to polyphonic music transcription. On the other hand, the nonnegative matrix factorization (NMF) is usually used for separating musical instrument sounds and extracting simple repeating patterns [27, 28] and only approximates complex audio mixture since the one-to-one mapping is uncertified. Efficient feature extraction from complex audio mixtures will be promising by combining lower-order analysis using structured models such as the HTC and higher-order analysis using unconstrained models such as the NMF.

6. Conclusions

We have described how musical genres of retrieved pieces shift by changing the volume of separated instrument parts and explained a QBE retrieval approach on the basis of such genre classification shift. This approach is important because it was not possible for a user to customize the QBE query in the past, which required the user to always find different pieces to obtain different retrieved results. By using the genre classification shift based on our original sound source separation method, it becomes easy and intuitive to customize the QBE query by simply changing the volume of instrument parts. Experimental results confirmed our hypothesis that the musical genre shifts in relation to the volume balance of instruments.

Although the current genre shift depends on only the volume balance, other factors such as rhythm patterns, sound effects, and chord progressions would also be useful for causing the shift if we could control them. In the future, we plan to pursue the promising approach proposed in this paper and develop a better QBE retrieval system that easily reflects the user's intention and preferences.

Appendix

Parameter Update Equations

The update equation for each parameter derived from the M-step of the EM algorithm is described here. We solved the simultaneous equations, that is, derivatives of the sum of the cost function (24), and Lagrange multipliers for model parameter constraints, (10) and (12), are equal to zero. Here we introduce the weighted sum of decomposed powers:

$$\begin{aligned} Z_{kl}(t,f) &= \alpha\Delta^{(J)}(k,l;t,f)X(t,f) + (1-\alpha)Y_{kl}(t,f), \\ Z^{(H)}_{klmn}(t,f) &= \Delta^{(H)}(m,n;k,l,t,f)Z_{kl}(t,f), \\ Z^{(I)}_{klmn}(t,f) &= \Delta^{(I)}(m,n;k,l,t,f)Z_{kl}(t,f). \end{aligned} \tag{A.1}$$

The summation or integration of the decomposed power over indices, variables, and suffixes is denoted by omitting these characters, for example,

$$Z_{kl}^{(H)}(t,f) = \sum_{m,n} Z_{klmn}^{(H)}(t,f),$$
$$Z_{klm}^{(H)}(t) = \sum_{n} \int Z_{klmn}^{(H)}(t,f)\,df. \tag{A.2}$$

$w_{kl}^{(J)}$ is the overall amplitude:

$$w_{kl}^{(J)} = Z_{kl}^{(H)} + Z_{kl}^{(I)}. \tag{A.3}$$

$w_{kl}^{(H)}$ and $w_{kl}^{(I)}$ are the relative amplitude of harmonic and inharmonic tone models:

$$w_{kl}^{(H)} = \frac{Z_{kl}^{(H)}}{Z_{kl}^{(H)} + Z_{kl}^{(I)}},$$
$$w_{kl}^{(I)} = \frac{Z_{kl}^{(I)}}{Z_{kl}^{(H)} + Z_{kl}^{(I)}}. \tag{A.4}$$

$u_{klm}^{(H)}$ is the amplitude coefficient of temporal power envelope for harmonic tone model:

$$u_{klm}^{(H)} = \frac{Z_{klm}^{(H)}}{Z_{kl}^{(H)}}. \tag{A.5}$$

$v_{kln}^{(H)}$ is the relative amplitude of the nth harmonic component:

$$v_{kln}^{(H)} = \frac{Z_{kln}^{(H)}}{Z_{kl}^{(H)}}. \tag{A.6}$$

$u_{kln}^{(I)}$ is the amplitude coefficient of temporal power envelope for inharmonic tone model:

$$u_{klm}^{(I)} = \frac{Z_{klm}^{(I)}}{Z_{kl}^{(I)}}. \tag{A.7}$$

$v_{kln}^{(I)}$ is the relative amplitude of the nth inharmonic component:

$$v_{kln}^{(I)} = \frac{Z_{kln}^{(I)}}{Z_{kl}^{(I)}}. \tag{A.8}$$

τ_{kl} is the onset time:

$$\tau_{kl} = \frac{\sum_{m} \int \left(t - m\rho_{kl}^{(H)}\right) Z_{klm}^{(H)}(t)\,dt + \sum_{m} \int \left(t - m\rho_{kl}^{(I)}\right) Z_{klm}^{(I)}(t)\,dt}{Z_{kl}^{(H)} + Z_{kl}^{(I)}} \tag{A.9}$$

$\omega_{kl}^{(H)}\omega_{kl}^{(H)}$ is the F0 of harmonic tone model:

$$\omega_{kl}^{(H)} = \frac{\sum_{n} \int n f Z_{kln}^{(H)}(f)\,df}{\sum_{n} n^2 Z_{kln}^{(H)}}, \tag{A.10}$$

$\sigma_{kl}^{(H)}$ is the diffusion of harmonic components along frequency axis:

$$\sigma_{kl}^{(H)} = \left(\frac{\sum_{n} \int \left(f - n\omega_{kl}^{(H)}\right)^2 Z_{kln}^{(H)}(f)\,df}{Z_{kl}^{(H)}} \right)^{1/2}. \tag{A.11}$$

Acknowledgments

This research was partially supported by the Ministry of Education, Science, Sports and Culture, a Grant-in-Aid for Scientific Research of Priority Areas, the Primordial Knowledge Model Core of Global COE program, and the JST CrestMuse Project.

References

[1] A. Rauber, E. Pampalk, and D. Merkl, "Using psycho-acoustic models and self-organizingmaps to create a hierarchical structuring of music bysound similarity," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '02)*, pp. 71–80, 2002.

[2] C. C. Yang, "The MACSIS acoustic indexingframework for music retrieval: an experimental study," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '02)*, pp. 53–62, 2002.

[3] E. Allamanche, J. Herre, O. Hellmuth, T. Kastner, and C. Ertel, "A multiple feature model for musical similarity retrieval," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '03)*, pp. 217–218, 2003.

[4] Y. Feng, Y. Zhuang, and Y. Pan, "Music information retrieval by detecting mood viacomputational media aesthetics," in *Proceedings of the International Conference on Web Intelligence (WI '03)*, pp. 235–241, 2003.

[5] B. Thoshkahna and K. R. Ramakrishnan, "Projektquebex: a query by example system for audioretrieval," in *Proceedings of the International Conference on Multimedia and Expo (ICME '05)*, pp. 265–268, 2005.

[6] F. Vignoli and S. Pauws, "A music retrievalsystem based on user-driven similarity and its evaluation," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '05)*, pp. 272–279, 2005.

[7] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Musical instrument recognizer "instrogram" and its application to music retrieval based on instrumentation similarity," in *Proceedings of the Annual International Supply Management Conference (ISM '06)*, pp. 265–274, 2006.

[8] L. Lu, D. Liu, and H. J. Zhang, "Automatic mood detection and tracking of music audio signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 5–18, 2006.

[9] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast features," in *Proceedings of the International Conference on Multimedia and Expo (ICME '02)*, pp. 113–116, 2002.

[10] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proceedings of the International Conference On Computer Vision (ICCV '98)*, pp. 59–66, 1998.

[11] T. Virtanen and A. Klapuri, "Separation of harmonic sounds using linear models for the overtone series," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP '02)*, vol. 2, pp. 1757–1760, 2002.

[12] M. R. Every and J. E. Szymanski, "A spectralfiltering approach to music signal separation," in *Proceedings of the Conference on Digital Audio Effects (DAFx '04)*, pp. 197–200, 2004.

[13] J. Woodruff, P. Pardo, and R. Dannenberg, "Remixing stereo music with score-informed source separation," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '06)*, pp. 314–319, 2006.

[14] H. Viste and G. Evangelista, "A method for separation of overlapping partials based on similarity of temporal envelopes in multichannel mixtures," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 3, pp. 1051–1061, 2006.

[15] D. Barry, D. Fitzgerald, E. Coyle, and B. Lawlor, "Drum source separation using percussive feature detection and spectral modulation," in *Proceedings of the Irish Signals and Systems Conference (ISSC '05)*, pp. 13–17, 2005.

[16] H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, T. Nishikawa, and K. Shikano, "Blind source separation combining independent component analysis and beamforming," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 11, pp. 1135–1146, 2003.

[17] M. A. Casey and A. Westner, "Separation of mixed audio sources by independent subspace analysis," in *Proceedings of the International Computer Music Conference (ICMC '00)*, pp. 154–161, 2000.

[18] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler, "Automatic music transcription and audio source separation," *Cybernetics and Systems*, vol. 33, no. 6, pp. 603–627, 2002.

[19] K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Integration and adaptation of harmonic and inharmonic models for separating polyphonic musical signals," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, pp. 57–60, 2007.

[20] H. Kameoka, T. Nishimoto, and S. Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 982–994, 2007.

[21] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: popular, classical, and jazz music databases," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '02)*, pp. 287–288, 2002.

[22] M. Goto, "AIST annotation for the RWC music database," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '06)*, pp. 359–360, 2006.

[23] R. J. Turetsky and D. P. W. Ellis, "Groundtruth transcriptions of real music from force-aligned MIDI synthesis," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '03)*, 2003.

[24] M. Muller, *Information Retrieval for Musicand Motion*, chapter 5, Springer, Berlin, Germany, 2007.

[25] N. Yasuraoka, T. Abe, K. Itoyama, K. Komatani, T. Ogata, and G. Hiroshi, "Changing timbre and phrase in existing musical performances as you like," in *Proceedings of the ACM International Conference on Multimedia (ACM-MM '09)*, pp. 203–212, 2009.

[26] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.

[27] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," in *Proceedings of the International Workshop on Independent Component Analysis and Signal Separation (ICA '06)*, pp. 700–707, April 2006.

[28] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 1–12, 2007.