

Research Article

Gamer's Facial Cloning for Online Interactive Games

Abdul Sattar,¹ Nicolas Stoiber,² Renaud Segulier,¹ and Gaspard Breton²

¹ SUPELEC/IETR, SCEE, Avenue de la Boulaie, 35576 Cesson-Sevigne, France

² Orange Labs, RD/TECH, 4 rue du Clos Courtel, 35510 Cesson-Sevigne, France

Correspondence should be addressed to Abdul Sattar, abdul.sattar@supelec.fr

Received 30 January 2009; Revised 3 June 2009; Accepted 11 July 2009

Recommended by Zhongke Wu

Virtual illustration of a human face is essential to enhance the mutual interaction in a cyber community. In this paper we propose a solution to solve two bottlenecks in facial analysis and synthesis for an interactive system of human face cloning for non-expert users of computer games. Tactical maneuvers of the gamer make single camera acquisition system unsuitable to analyze and track the face due to its large lateral movements. For an improved facial analysis system, we propose to acquire the facial images from multiple cameras and analyze them by multiobjective 2.5D Active Appearance Model (MOAAM). Facial morphological dissimilarities between a human face and an avatar make the facial synthesis quite complex. To successfully clone or retarget the gamer facial expressions and gestures on to an avatar, we introduce a simple mathematical link between their appearances. Results obtained validate the efficiency, accuracy and robustness achieved.

Copyright © 2009 Abdul Sattar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Over the last decade computer games have become more and more an interactive entertainment. Virtual representation of a character has gained the interest of both gamers and researchers. Gamers do not want to sit and play the games, instead they need to get involved in the game to an extent to visualize opponent's face and interact with him virtually. The use of virtual representation of a human face in game consoles or creating avatars has been tremendously increasing. In addition, a growing number of websites now host virtual characters technologies to deliver their contents in a more natural and friendly manner. Gestures and features (e.g., eyes, nose, mouth and eyebrows) of a human face are actually the reflection of a person's inner emotional state and personality. They are also believed to play an important role in social interactions, as they give clues to a gamer's state of mind and therefore help the communication partner to sense the tone of a speech, or the meaning of a particular behavior. For these reasons, they can be identified as an essential nonverbal communication channel in game consoles.

To track, analyze and synthesize gamer's face efficiently and to ensure the interaction of a gamer, system needs to overcome two bottlenecks in facial analysis and synthesis. Facial analysis deals with the face alignment, pose, features,

gestures and emotions extractions. Excitements caused by the tactical moves of a game, compel the gamer to move around in various directions. These maneuvers produce large lateral movements of a face, which makes it difficult for a facial analysis system to track and analyze the face. For a facial synthesis system, cloning or retargeting the features, emotions and orientation of a human face on to an avatar is again one of the challenging tasks. Cloning or retargeting is difficult due to the facial morphological differences between a real face and an avatar. Furthermore, large and complex face deformations due to the expressions made by a nonrigid human face makes the online system computationally complex to clone or replicate it on to an avatar.

We propose a robust and efficient gamer's online cloning interactive system as shown in Figure 1. Our system is composed of two cameras installed on the extreme edges of the screen to acquire real-time images of the gamer. Gamer's face is analyzed and his pose and expressions are synthesized by the system to clone or retarget his features in the form of an avatar so that the gamers can interact with each other virtually. In the following paragraphs we briefly explain solutions by the facial analysis and synthesis systems, embedded in our proposed interactive system.

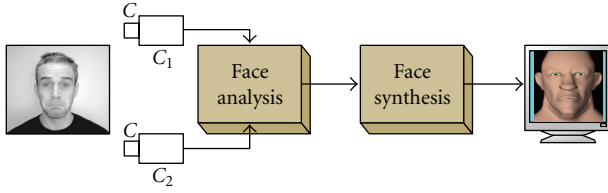


FIGURE 1: Global system.

1.1. Face Analysis. Human faces are nonrigid objects. The flexibility of a face is well tackled with the appearance-based or deformable model methods [1], which are remarkably efficient for features extraction and alignment of frontal-view faces. As we will see in Section 2, researchers worked out the bottlenecks of face analysis by emphasizing on the model generation and their search methodologies. However we emphasize on increasing the amount of data to be processed with the help of multiple cameras as shown in Figure 1. In single-view system face alignment cannot be accomplished when a face occludes itself during its lateral motion, such as in a profile view only half of the face is visible. To overcome this dilemma we exploit data from another camera and associate it with the one unable to analyze at the first place. In multicamera system, optimization of more than one error is to be performed between a model and query images from each camera. Searching for an optimum solution of a single task employing two or more distinct errors requires multiobjective optimization (MOO). Many MOO techniques exist but to analyze the face we propose optimization of MOAAM by Pareto-based NSGA-II [2] due to its exploitation and exploration ability, nondominating strategy and population based approach which provide the mutual interaction of the results by multiple cameras. In this paper, we use our previous work of [3] and improved our system by obtaining new results based on a new synthetic face database.

1.2. Face Synthesis. In facial synthesis system the purpose is to retarget or clone gamer's face orientation and its features on the synthetic model so that the gamers can interact with each other virtually. Cloning and retargeting is difficult, because avatar does not have the same morphology as the gamer. Our contribution in this system is the introduction of a simple mathematical relation between their appearances called ATM (Appearance Transformation Matrix). To calculate it we make use of two databases explained in Section 5.1. The first database is a large collection of human facial expressions (H-database) and the second database is an optimal database of synthetic facial expressions (A-database) constructed for the avatar based on the analysis of the H-database. Our second contribution is to provide an interactive system for the gamer to build his own database and calculate gamer's specific ATM. The generation of the gamer's database is based on our face analysis system of MOAAM and is obtained by requesting the gamer to imitate few specific and relevant facial expressions displayed on the screen.

Whole system works in two phases. First of all, user's oriented face is analysed by MOAAM, which gives its appearance and pose parameters. These appearance parameters are pose-free and belongs to the frontal face of the user. Therefore they are transformed by ATM in the synthetic face's parameter space and synthetic face is synthesized accordingly. After that pose parameters obtained previously by MOAAM analysis are used to adjust the orientation of the avatar being displayed on the screen.

Remaining of the paper is organized as follow. Section 2 presents the previous and related work in both the domains of facial analysis and synthesis. Section 3 presents the preliminary concepts of our system. Section 4 describes the work done in face analysis. Section 5 explains the system to synthesize a face. Detailed description of our proposed interactive system is elaborated in Section 6, while Section 7 concludes the paper.

2. Previous and Related Work

In this section we have divided the previous and related work for both facial analysis and synthesis into two subsections. However, our first contribution in the facial analysis domain is explained in detail in Section 4. And our second contribution in the facial synthesis domain is explained in Section 5.

2.1. Face Analysis

2.1.1. Multiple 2DAAM. Active Appearance Model (AAM) is one of the well-known deformable method [1] efficient in feature extraction and alignment of a face. References [4, 5] performed pose prediction by using 3 AAM models, one dedicated to the frontal view and two for the profile views. References [6, 7] implemented Active Shape Model (ASM) for the face alignment, by using 5 poses of each face to create a model. Reference [8] also used 3 DAMs (Direct Appearance Models) for face alignment. Reference [9] used another appearance based architecture employing 5 view-specific template detectors to track large range head yaw by a monocular camera. The Radial Basis Function Network interpolates the response vectors obtained from normalized correlation from the input image and 5 template detectors.

Use of more than one model of AAM has some disadvantages: (i) Storage of shapes and textures of the images of all the models requires an enormous amount of storage memory. (ii) Extensive processing of computing 3 AAM in parallel to determine the model required for query images, eventually makes the system sluggish. Moreover classical AAM search methodology requires precomputed regression matrices, which become a burden on time and memory as the amount of training images increases.

Coupled View AAM is used in [10] to estimate the pose. In the training phase they include 2D shapes and 2D textures of both frontal and profile views of each subject. Appearance parameters of their CV-AAM have the capability to estimate the pose. Appearance parameters of their model can tune both the shape and the profile angle of a face. For the profile angle estimation they have used several appearance

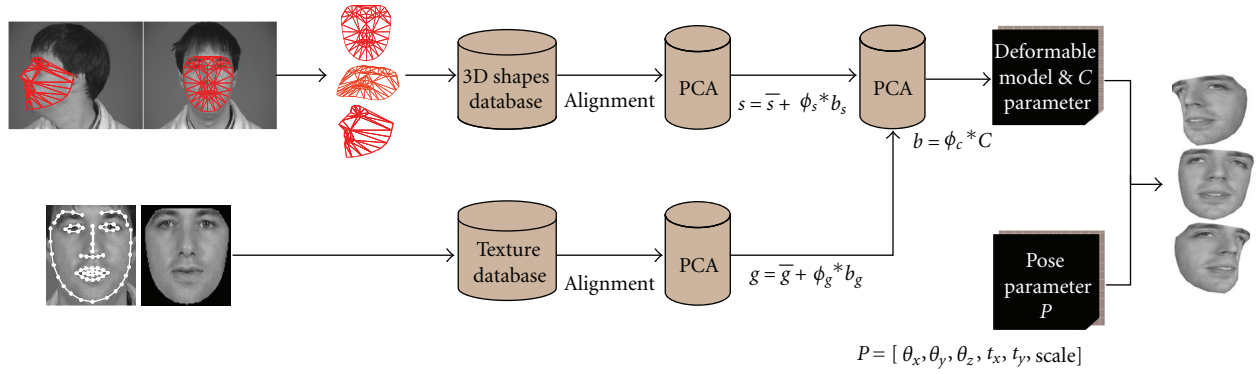


FIGURE 2: AAM modeling.

parameters which can be replaced by one pose parameter in a 3D AAM. Thus, increase in the number of parameters decreases the rapidness of the system.

2.1.2. 3DAAM. Face can also be aligned by 3D deformable model methods in which a set of images are annotated in 3D to model a face. Reference [11] used 3D face model Candide along with simple gradient descent method as a search algorithm for face tracking. References [12] used 2D+3D AAM along with a fitting algorithm, called inverse compositional image alignment algorithm, which is again an extension of a gradient descent method. Reference [13] applied 3D AAM for face tracking in a video sequence using same IC-LK (Inverse Compositional Lucas-Kanade) algorithm. The optimization by gradient descent lack the properties of exploration and diversity, hence cannot be used in MOO. In our previous work of [14] we have used genetic algorithm instead of gradient descent for the optimization in 2.5D AAM.

2.1.3. Multiview Fitting by 2D or 3DAAM. Pose angles can be estimated by fitting the above 2D or 3D deformable models on multiple images acquired by two, three or multiple cameras. Reference [15] proposed a robust algorithm of fitting a 2D+3D AAM to multiple images acquired at the same instance. Their fitting methodology, instead of decomposing into three independent optimizations from three cameras, adds all the errors. Moreover they used gradient descent (ICLK: Inverse Compositional Lukas Kanade) algorithm as a fitting method, which eventually requires to precompute Jacobians and Hessian matrix. Reference [16] proposed another algorithm of face tracking by Stereo Active Appearance Model (STAAM) fitting, which is an extension of the above fitting of 2D+3D AAM to multiple images. Lack of exploration capability of the method makes ICLK very sensitive to initialization.

In [17] the advantages of adaptive appearance model based method with a 3D data-based tracker using sparse stereo data is combined. Reference [18] proposed a model-based stereo head tracking algorithm and is able to track six degrees of freedom of head motions. Their face model contains 300 triangles compare to our 113 triangles usually

used in classical AAM and ICLK based AAM and so forth. Moreover their initialization process requires user intervention. Reference [19] performed 2D head tracking for each subject from multiple cameras and obtained 3D head coordinates by triangulation. Lack of ground truth error calculations creates uncertainty in the accuracy of their system. Furthermore slight calibration error massively deteriorates the triangulation.

Our proposition of face alignment is based on two cameras using 2.5D AAM optimized by Pareto-based multiobjective genetic optimization of NSGA-II. It not only eliminates the steps of precomputation but also provides both exploration and exploitation capability in the search by NSGA-II. Hence it is not sensitive to initialization.

2.2. Face Synthesis. By facial cloning, we refer to the action of transferring the animation from a source (typically a human face) to a target (another human face or a synthetic one). The cloning (or *retargeting*) can be either direct or indirect. In direct retargeting, the purpose is to transfer the motion itself of a few selected interest markers (and optionally a texture) from one face to another [20]. The marker trajectories usually undergo a transformation that compensates for the morphological differences between the source and the target face [21–24]. This morphological adaptation is not always satisfactory, especially if the source and the target faces are very different. An interesting way to get around this difficulty is to turn to indirect retargeting. In indirect retargeting, the motion data is not transferred as such, but is first converted by a specific model to a better representation space, or parameter space, more suited for the motion transfer [25–27]. In the next paragraph we will go over some of the most common representations used for indirect retargeting.

In order for a facial parameterization to be suited for retargeting applications, it must be adapted to the extraction of parameters from motion capture data, and offer an accurate description of facial deformations. Early parameterization schemes like direct parameterizations [28] or pseudomuscle systems [29–31] usually have the advantage of being simple to conceptualize and computationally efficient, but the obtained parameter sets are generally not optimal. In particular, when not operated carefully, they

can generate inconsistent facial configurations. Besides, it is not straightforward to extract the values of the parameters from raw facial motion data (video or 3D motion capture). Muscle physics systems attempt to simulate more rigorously the mechanical behavior of the human face, and thus tend to improve the degree of realism of facial deformations [32]. Yet, as for direct parameterization, the manipulation of the muscle network is not particularly intuitive, and the extraction of muscular contractions from video or motion capture data remains an open problem [33]. A popular facial parameterization which directly originates from observation is the Facial Action Coding System (FACS) [34]. This scheme was originally meant to describe facial expressions in a standardized way in terms of combination of basic facial Action Units (AU). Its coherence and good practical performances made it an interesting tool on which to build performance based animation systems. The MPEG-4 standard later extended this concept for facial animation compression purposes, introducing the Facial Animation Parameters (FAP) [35]. The FACS and MPEG-4 FAP have been used to capture and retarget static and dynamic facial expressions between human and synthetic faces [36, 37]. The disadvantage of methods based on multiple separate action units, is that the natural correlation between multiple facial action occurring in each facial expression is ignored. Thus the animation resulting from these approaches tend to be somewhat nonhuman or robotic.

More recently studies have aimed at obtaining more natural parameterization by performing a statistical modeling of the facial motion. This consist in gathering a collection of relevant examples (database) and to statistically detect particular variation modes, which encompass the specificity of the source or the target. The facial parameters correspond to the contribution of these modes. When two faces have corresponding models, Animations can be easily transferred by mapping the model parameters from one face to the other. Many studies have pointed that motion data consisting of only the positions of a few markers cannot efficiently capture the subtleties of human facial expressions, and have proposed to also capture the textural information [38]. Active Appearance Models (AAM) are frequently used for that propose, since they encompass the motion of well chosen geometric points as well as the pixel intensity changes occurring on the faces, which account for finer deformation of the skin [1]. References [39, 40] obtain impressive results of facial expressions transfer between multiple human faces based on an AAM parameterization. For this type of retargeting scheme to be successful however, the appearance models of the source and the target must characterize the same scope of expressions. In particular their databases must correspond. Constructing a database of expressions for a synthetic face which matches the scope of the source human database is not trivial. Reference [41] transfer facial expressions from the AAM parameters of a human face to an avatar based on a blendshape database. The database of the avatar consists of key expressions selected from the human database, however too few expressions are used for the virtual face to allow for a detailed expression retargeting. Reference [42] later improved this approach by

preprocessing the human database in order to automatically isolate individual facial actions. Each of the facial actions can then be reproduced on the avatar to construct a blendshape database. For a reasonable number of facial expressions, this approach ensures the compatibility between the source and target database, without requiring the construction of many avatar facial examples. Yet, for a more complete scope of facial movements, the number of individual facial actions can become large, and thus the number of facial configurations for the avatar database as well. Moreover, by decomposing the expressions into individual units, the correlation between these units when performing an expression is lost in the parameterization. Reference [27] performs a linear retargeting of monocular human appearance parameters to muscle-based animation parameters. The transfer function is based on the matching of a human database of key expressions with a database of corresponding animation parameters for the synthetic face. Yet, the choice of the database key expressions is subjective in that case. Moreover the synthetic face is animated with muscle contraction parameters which can sometimes lead to incoherent interpolation, and prevents the system from being used with other types of animation methods.

We propose a new method to efficiently transfer facial expressions from a human face to a synthetic face, based on pose-free active appearance model parameters delivered by our multiple camera system. The method analyzes a human expression database, and automatically determines which key expressions have to be constructed in the avatar database for the expression retargeting to be efficient.

3. Preliminary Concepts

3.1. 2.5D AAM Modeling. 2.5D AAM of [3, 14] is constructed by (i) 2D landmarks of the frontal view (width and height of a face model) and x coordinates of landmarks in profile view (depth of a face model) combined to make 3D shape model and (ii) 2D texture of only frontal view mapped on its 3D shape. In the training phase of 2.5D AAM, 68 points are marked manually as shown in Figure 2.

All the landmarks obtained previously are resized and aligned in three dimensions using Procrustes analysis ([43, 44]). The mean of these 3D landmarks is calculated which is called mean shape. Principal Component Analysis (PCA) is performed on these shapes to obtain shape parameters with 95% of the variation stored in them:

$$s_i = \bar{s} + \phi_s * b_s, \quad (1)$$

where s_i is the synthesized shape, \bar{s} is the mean shape, ϕ_s are the eigenvectors obtained during PCA and b_s are the shape parameters.

The 3D mean shape obtained in the previous step is used to extract and warp (based on the Delaunay triangulation) the frontal views of all the face images. Only two dimensions of the mean shape are used to get 2D frontal view textures. That is why we call our model as 2.5D AAM, since it is composed of landmarks represented in 3D domain but only 2D texture is warped on this shape to adapt 2.5D model.



FIGURE 3: Snapshots of rotating 2.5D AAM.

Mean of these textures is calculated. Followed by, another PCA to acquire texture parameters with 95% of the variation stored in these parameters:

$$g_i = \bar{g} + \phi_g * b_g, \quad (2)$$

where g_i is the synthesized texture, \bar{g} is the mean texture, ϕ_g are the eigenvectors obtained during PCA and b_g are the texture parameters.

Both of the above parameters are combined by concatenation of b_s and b_g . And a final PCA is performed to obtain the appearance parameters:

$$b = [b_s, b_g]^T, \quad b = \phi_C * C, \quad (3)$$

where ϕ_C are the eigenvectors obtained by retaining 95% of the variation and C is the matrix of the appearance parameters, which are used to obtain shape and texture of each face of the database.

2.5D model can be translated as well as rotated with the help of pose vector P :

$$P = [\theta_x, \theta_y, \theta_z, t_x, t_y, \text{Scale}]^T, \quad (4)$$

where θ_x corresponds to the face rotating around the x axis (pitch: shaking head up and down), θ_y to the face rotating around the y axis (yaw: profile views) and θ_z to the face rotating around the z axis (roll). t_x, t_y are the offset values from the supposed origin and Scale is a scalar value for the magnification of the model in all the dimensions. Figure 3 shows the model rotating by changing θ_y , making left and right semi profile views.

In segmentation this deformed, rotated and translated shape model obtained by varying C and P parameters, is placed on the query image I to warp the face to mean frontal shape. After this shape normalization we apply photometric texture normalization to overcome illumination variations. The objective is to minimize pixel error

$$e = \sqrt{\sum_x [I(C, P) - M(C)]^2}, \quad (5)$$

where $I(C, P)$ is the segmented image and $M(C)$ is the model obtained by C parameters. To choose good parameters we need an optimization method. In our proposition, both of these pose P and appearance parameters C are optimized by genetic optimization of NSGA-II.

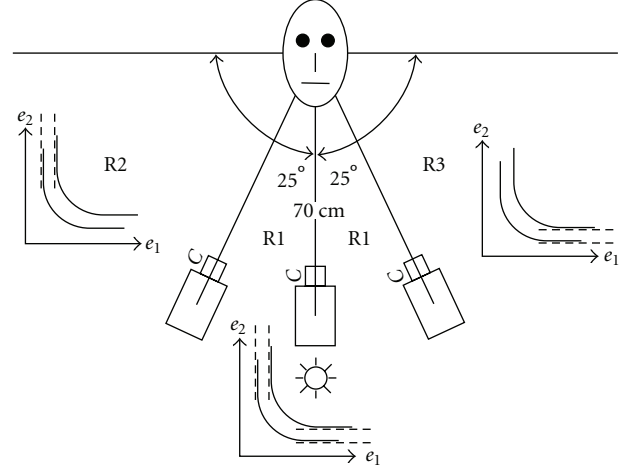


FIGURE 4: MultiView System.

3.2. Multiple Camera System. In single-view system face alignment cannot be accomplished when a face occludes itself during its lateral motion. Such as in a profile view only half of the face is visible. To overcome this dilemma we exploit data from another camera and associate it with the one unable to analyze at the first place. This association helps the search methodology to reduce the possibility of divergence. Moreover better outcomes of one camera can escort the other. In multiview systems, higher the amount of processing data higher is the robustness ability of a system however efficiency deteriorates due to high consumption of processing time and memory. In other words a trade-off is required between robustness and efficiency.

A database of facial images capable of self assessing is desired to validate our application. The community lacks such a database which involves lateral motion of a face captured by more than one camera. In order to implement our application we developed a multiview scenario. The purpose of constructing this multiview system is to emulate the scenario of integrating two off the shelf webcams placed on the extreme edges of the display screen facing towards the user as shown in Figure 4.

AAM rendered on the facial images of both webcams are blended together to represent a face model seen by a virtual camera placed in between. The results of this virtual webcam are compared by a third camera actually placed at the center. In other words it is a comparison between a multicamera system by MOAAM with single-camera system by SOAAM (Single-Objective AAM). These three cameras are placed 25 degrees apart on a boundary of a circle with a radius of 70 cm as shown in Figure 4. Center of this circle serves as a principal point for each camera. Seven individuals from a research team are invited for screen shots with the intention of obtaining 1218 images with lateral motion. Each individual rotates his face gradually from frontal view to left and right profile views. At each instance three images from each webcam are acquired simultaneously to obtain temporally synchronized images.



FIGURE 5: Test database images: Same pose from 3 webcams.

3.2.1. Illumination. It remains steady through out the sequence. It is accomplished by a white ambient light placed behind the central camera as shown in the Figure 4. The light we used comes with the stand and a built-in umbrella holder to give extra flexibility. By adjusting the umbrella's position we have rejected the bright spot on the face. It works well for taking facial images with webcams.

3.2.2. Camera Calibration. It is performed by a publicly available toolbox [45]. A simple planar checkerboard is placed in front of the cameras and sequence of images are taken to calculate calibration parameters. With the help of the toolbox, four corners of the checker board are extracted and calibration is performed with respect to the grid of the checkerboard. The toolbox calculates intrinsic parameters (focal length, principal point, distortion and skew) and extrinsic parameters (rotation vector and translation vector) for each camera. With the help of these parameters, all the facial images of these cameras are calibrated.

Figure 5 shows some images of test database acquired from three webcams. A similar scenario is emulated in the software MAYA for a video of synthetic faces. The synthetic face database does not contain camera calibration error hence it is helpful to analyze results free of calibration errors. Figure 6 show some examples of test database of synthetic faces (Synthetic face in the first row was obtained from www.ballistic.com, while remaining face models were made in a software named as "Facial Studio". All of them were imported in MAYA for rendering the synthetic facial images). Some of the facial images of M2VTS [46] (learning database) are also shown in Figure 7.

4. Face Analysis

The main objective of our application is to clone a real human face in the form of an avatar. For such an application face analysis plays an important role for face synthesis. The more efficient the analysis is, facial synthesis is likely to be more accurate. To obtain an efficient and robust face analysis system we acquire a human face with two cameras and analyze it by an appearance based morphable model of 2.5D AAM.

4.1. MOAAM. In single-view system, single error between model and query image is optimized. However in multiview system, the optimization of more than one error is to be performed between a model and query images from each camera. AAM fitting on multiviews is shown in Figure 8. In multiview AAM, the model is rendered on both the images from each camera with the same C parameters.

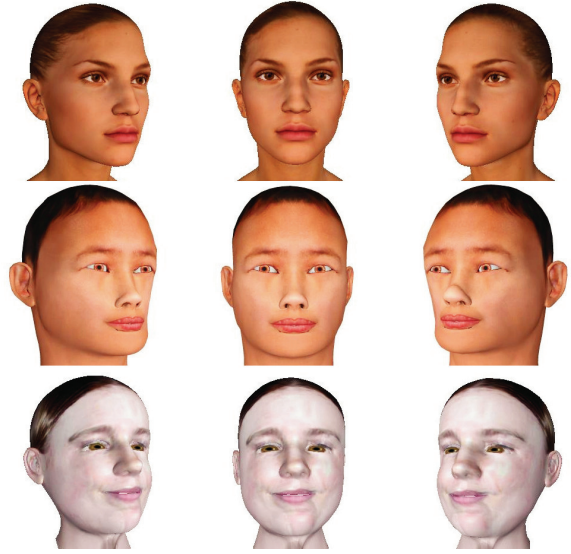


FIGURE 6: Test database synthetic images.



FIGURE 7: Learning database images.

The P parameters also remain the same except a yaw angle offset (θ_{offset}) is introduced between the models rendering on two images. After segmentation, pixel errors between both the images and models are calculated. The objective is to minimize pixel error of (5) obtained from each of the two cameras

$$\begin{aligned} e_1 &= \sqrt{\sum_x [I_1(C, P_1) - M(C)]^2}, \\ e_2 &= \sqrt{\sum_x [I_2(C, P_2) - M(C)]^2}, \end{aligned} \quad (6)$$

where P_1 and P_2 are linked by an offset of yaw angle. In order to optimize both errors we propose Pareto-based NSGA-II MOO.

4.1.1. NSGA-II. Genetic Algorithm is a well-known search technique. We have used its multiobjective version of Non-dominated Sorting Genetic Algorithm (NSGA-II) proposed by [2] to optimize the appearance C and pose parameters P . The target is to find out the best possible values of these parameters giving minimum pixel errors between the model and the query images of both cameras. In this optimization technique each parameter is considered as a gene. All the genes of C and P are concatenated to form a chromosome. A population of particular number of chromosomes is randomly created. Pixel errors (fitness) between query images and the model (represented by each chromosome) are calculated. Tournament selection is applied to select

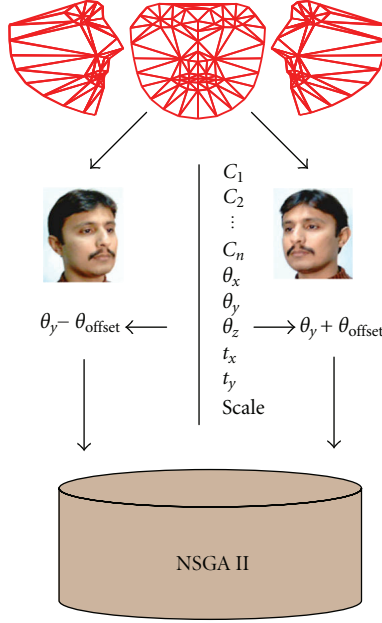


FIGURE 8: Fitting of MOAAM.

parents from the population to undergo reproduction. Two point crossover and Gaussian mutation is implemented to reproduce the next generation of chromosomes. Selection and reproduction is based upon nondominating sort. The objective is to minimize both of these pixel errors, hence nondominating scenario is to be implemented by Pareto optimization.

4.1.2. Pareto Fronts. The fitting of AAM to image data is performed by minimization of the error function. In MOO several error functions are to be minimized, hence mutual relation of these errors point towards the appropriate MOO method. Dominating errors can be dealt with non Pareto-based MOO, but in this scenario both cameras serves the same purpose of acquiring images of a face. Hence non-dominating scenario is to be implemented with the desired Pareto optimum solution. The basic idea is to find the set of solutions in the population that are Pareto nondominated by the rest of the population as shown in Figure 9(a). These solutions are assigned the highest rank and are removed from further assignment of the ranks. Similarly, the remaining population undergoes the same process of ranking until the population is suitably ranked in the form of Pareto fronts as shown in the Figure 9(b). In this process some kind of diversity is required in the solutions to avoid convergence to a single point on the front. This diversity can be achieved by the exploration quality of Genetic Algorithm.

4.1.3. Switching of MOO to SOO. Processing data from two cameras is meaningful as long as they are relevant. With respect to a camera if a face is oriented such a way that it occludes itself there is no need of processing data from this camera. Eventually in order to avoid wastage of processing we divide field of views of both cameras in three regions R1,

R2 and R3 as shown in Figure 4. To determine the region of the face orientation Pareto-based NSGA-II is applied to evolve populations until small number of generations. After each generation evolution, the histogram of genes of the entire population representing the yaw of a face is observed. This histogram follows one of the three curves of Figure 10. Histogram curve-1 corresponds to region-1, where the information from both the cameras are meaningful and data from any one of them cannot be neglected. Whereas histogram curve-2 and curve-3 corresponds to region-2 and region-3 respectively, where the information from one of the camera is sufficient enough to localize the facial features and other camera can be discarded. After few generations, current population decides whether to stay in MOO or to switch to single objective optimization (SOO). Mathematically, let us suppose Pop is a set of population given as

$$Pop = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1k} & \cdots & X_{1M} \\ X_{21} & X_{22} & \cdots & X_{2k} & \cdots & X_{2M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{N1} & X_{N2} & \cdots & X_{Nk} & \cdots & X_{NM} \end{bmatrix}, \quad (7)$$

where N is the number of chromosomes X and M is the number of genes of each chromosome. Now we observe the k_{th} gene of each chromosome which represents yaw angle of the model. In order to calculate the histogram of chromosomes, we assign 1 to ζ such as

$$\zeta_i = \begin{cases} 1 & -\theta_{th} \leq X_{ik} \leq \theta_{th} \\ 0 & X_{ik} \leq -\theta_{th} \text{ or } X_{ik} \geq \theta_{th} \end{cases} \quad 1 \leq i \leq N, \quad (8)$$

where θ_{th} is the threshold angle equals to the half of the angle between two cameras. ϵ is the ratio of number of chromosomes representing the face position in region-1 to the total number of chromosomes:

$$\epsilon = \frac{\sum_{i=1}^N \zeta_i}{N} = \begin{cases} < 0.50, & \text{Single camera mode,} \\ \geq 0.50, & \text{Multiview mode.} \end{cases} \quad (9)$$

The value of ϵ decides whether to stay in MOO and utilize both cameras or to switch to single camera mode.

4.1.4. MOAAM Fitting. For MOAAM (also called MVAAM: Multiview AAM) fitting we refer readers to our previous work of [3], which illustrates stepwise detailed description of MOAAM fitting on a query image. It includes steps of initialization, reproduction, segmentation, fitness calculations, nondominating sort, replacement and switching of MOO to SOO. In our previous work we have highlighted the effects of slight errors caused by the camera calibration and the ground truth points for a real face database.

Camera calibration problem arises when we compare MOAAM results to SOAAM. As we have already mentioned in Section 3.2 that models obtained from two cameras placed

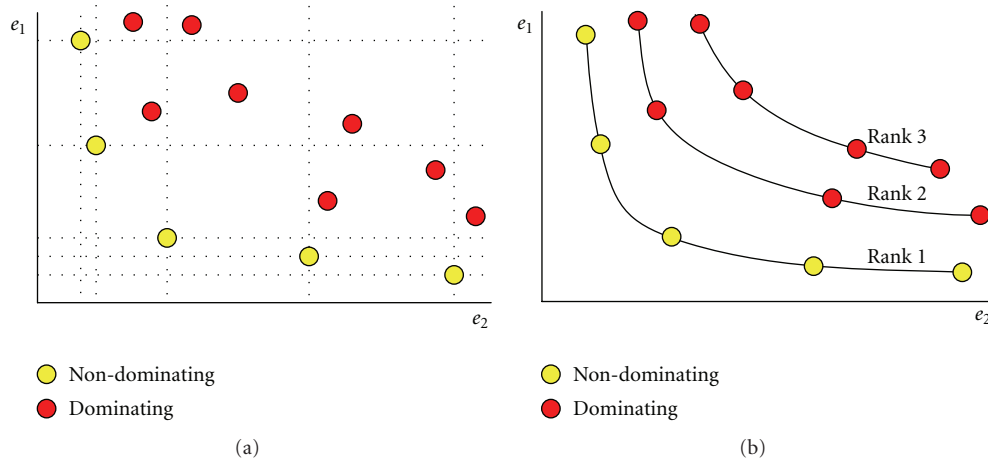


FIGURE 9: Pareto Fronts.

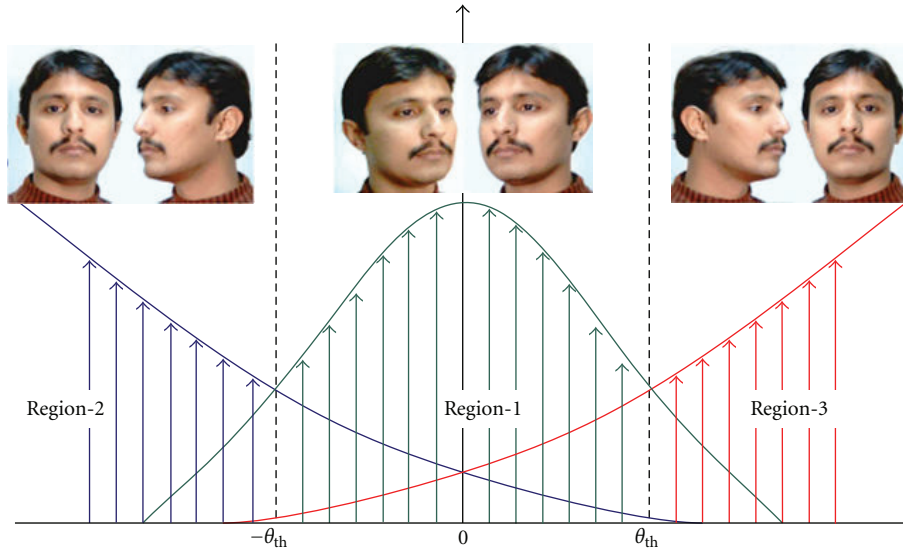


FIGURE 10: Histogram of chromosomes versus head orientation.

at the extreme edges of the display are blended together to compare it with the one obtained from the central camera. This comparison is highly prone to the calibration error of all the three cameras. Whereas the results from a single camera (SOAAM) do not experience any calibration problem. In this paper we have managed to overcome this dilemma by building a synthetic face database of several individuals. The scenario shown in Figure 4 is emulated, in the software named as MAYA, by placing different synthetic characters in between two virtual cameras each calibrated and located 50° apart. A third camera is placed in-between these two cameras for the comparison of results of a single camera and double camera. These cameras have all the characteristics of an actual camera along with the capability to fix intrinsic and extrinsic parameters to obtain 100% calibration.

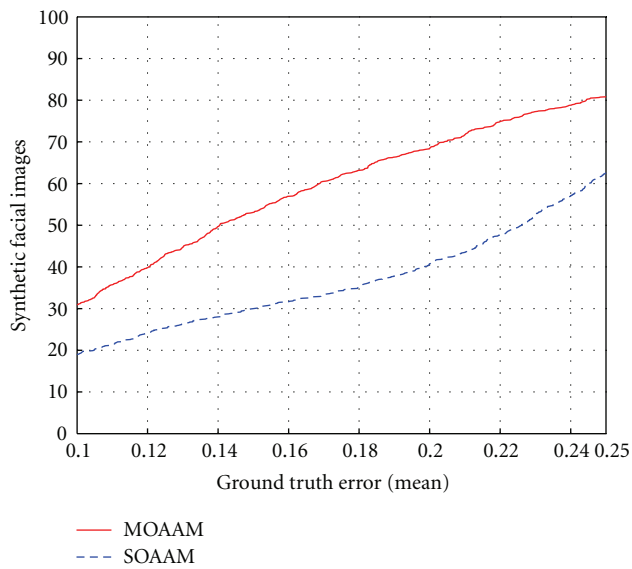
Ground truth points are the exact localization of the face orientation and features (nose, eyes and mouth). In

real face database there is a possibility of slight errors in the ground truth points since they are marked manually on each facial feature of each image. However in synthetic facial images this problem is solved by obtaining these locations automatically through scripts written in MAYA. With all these modifications we have verified our proposition of MOAAM and have updated our results.

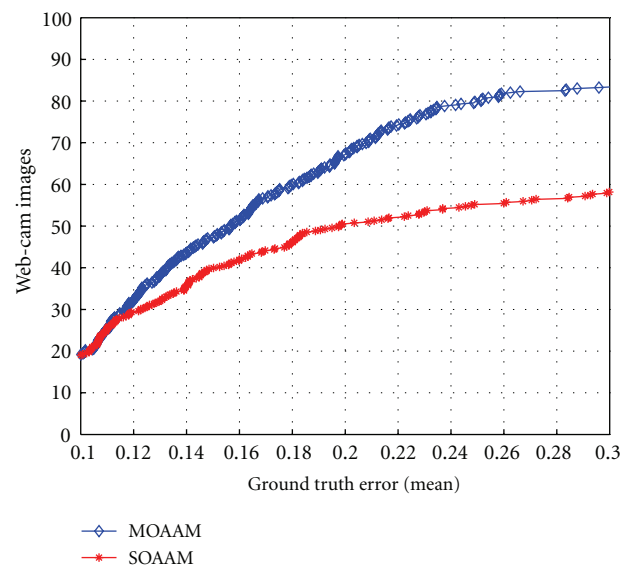
4.2. Experimental Results. We performed simulations using 64×64 pixels AAM by annotating 37 subjects of publicly available databases of M2VTS [46]. However for testing database we have used both real face database and synthetic face database. Both these databases contain 2418 facial images, of 7 real and 10 synthetic faces, from each camera. Among 2418, 806 images are considered to be taken from central camera to validate our results. In testing phase face alignment is performed on all the views from left profile



FIGURE 11: (a) and (b) Comparison of SOAAM and MOAAM (operating in R2 or R3). (c) and (d) Comparison of SOAAM and MOAAM (operating in R1).



(a)



(b)

FIGURE 12: (a) Comparison of GTE_{mean} for MOAAM and SOAAM (Synthetic face images). (b) Comparison of GTE_{mean} for MOAAM and SOAAM (Web-cam images).

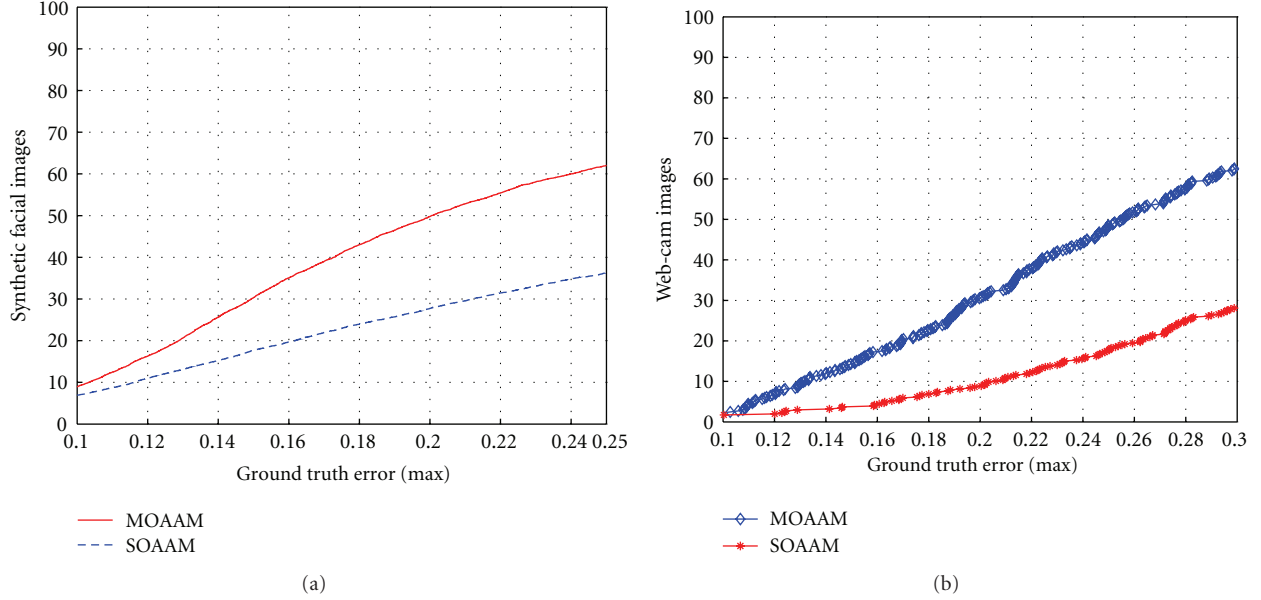


FIGURE 13: (a) Comparison of GTE_{max} for MOAAM and SOAAM (Synthetic face images). (b) Comparison of GTE_{max} for MOAAM and SOAAM (Web-cam images).

to right profile. Two sets of experiments are performed: SOAAM and MOAAM.

4.2.1. Single-Objective AAM. In SOAAM, AAM is rendered on the image sequence from the central camera, which is placed to highlight the benefit of MOAAM. As far as optimization is concerned, SOAAM is optimized by classical GA optimization. Same selection and reproduction criteria of NSGA-II are implemented in GA, in order to give a good comparison.

4.2.2. Multiobjective AAM. In MOAAM, same AAM is rendered on the face image sequence from the other two cameras, which are actually the part of our multiview system. Localization of face on these two images from each camera is performed by Pareto-based MOO of NSGA-II.

Best chromosomes obtained at the end of MOAAM and SOAAM contain best appearance and pose parameters for a given face. Features like eyes, nose and mouth can be extracted from these shapes as shown in Figure 11. First three rows correspond to synthetic faces while remaining rows represent real human faces. It can be seen from the images that as the face moves laterally the feature localization gets far better in two cameras (MOAAM) than in single central camera (SOAAM).

Figure 12(a) shows percentage of aligned synthetic images versus mean ground truth error (GTE_{mean}) of facial features (eyes, nose and mouth). GTE_{mean} is actually the mean error obtained by comparing MOAAM analyzed locations and manually marked locations of all the facial features of a facial image. The error is normalized by D_{eye} which corresponds to the distance between eyes, that is, an error of 1 corresponds to a mean error equal to the distance between the eyes. To eliminate the vagueness of ground truth

markings we consider results starting from 0.1 of D_{eye} , which means any two algorithms having a GTE_{mean} less than 0.1 is considered to be equally accurate. While for the maximum threshold results less than 0.25 of D_{eye} is considered to be well converged results. Figure 12(a) depicts that our system of MOAAM fitting by NSGA-II is a lot better than SOAAM fitting. In MOAAM 69% of the images are aligned with a GTE_{mean} less than 0.2 of D_{eye} . Whereas SOAAM aligned 41% of the total images. Similarly Figure 12(b) shows the results of experiments on real faces (previous work); MOAAM 68% and SOAAM 50%.

Figures 13(a) and 13(b) illustrate the comparison of both algorithms with respect to normalized maximum ground truth error (GTE_{max}) for both synthetic and real facial images databases respectively. GTE_{max} represents worst localization of a facial feature (eyes, nose or mouth) normalized by D_{eye} . Figure 13(a) depicts that MOAAM aligned 50% and SOAAM aligned 28% of synthetic facial images with GTE_{max} less than 0.2 of D_{eye} . Whereas Figure 13(b) shows MOAAM aligned 30% and SOAAM aligned 10% of real faces.

As far as time consumption is concerned, it is obvious that at the worst MOAAM required twice of the processing time compared to SOAAM but at the same time accuracy, robustness and increased field of view (FOV) is achieved. Moreover our technique of finding the region of face and discarding the data from the camera by NSGA-II reduces this twice factor. SOAAM required 1600 warps whereas MOAAM instead of 3200 warps required 2700 warps. Each warp equals 90% of the time consumed by an iteration, that is, 0.03 milliseconds in Pentium-IV 3.2 GHz. Therefore each facial image requires 90 milliseconds for the analysis without any prior knowledge of the pose, however in tracking mode we can reduce this time by employing pose parameters of previous frames, which eventually reduces the number of

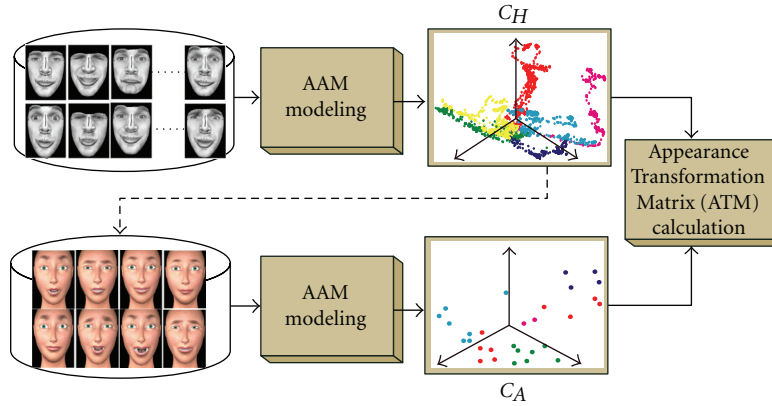


FIGURE 14: Overview of the face synthesis system. (Colors represent different types of expressions and are shown for the clarity of the display only).

warps (iterations). Moreover facial analysis by MOAAM can be made as a generic or a person specific MOAAM. In generic MOAAM the query face is totally unknown and to analyze it we need a vast learning database, whereas in person specific MOAAM model is generated from facial images of the same individual who would be analyzed by the system. Eventually person specific MOAAM is more time efficient and robust compared to generic MOAAM.

5. Face Synthesis

The goal of our application is to clone the gamer's facial expression to an avatar. The cloning consists of transferring the facial expressions from a source (typically a human face) to a target (another human face or a synthetic one). The avatar facial deformations then originate from real human movements (performance-based facial animation), which usually look more natural than manually-designed facial animation. Moreover, since the expressions of the gamer are captured and transferred in real-time, the facial animation of the avatar acts as a real gaming experience, and significantly improves the interactivity of the game compared to prerecorded animation sequences.

5.1. System Description. In this section, we present a general description of a system that provides an efficient parameterization of an avatars face for the production of emotional facial expressions, relying on captured human facial data. Here we make use of two databases of our previous work of [47]. An illustration of the system and its applications is displayed on Figure 14.

5.1.1. H-Database. The entry point of the system is a database of approximately 4000 facial images of emotional expressions (H-database). These images have been acquired on an actor performing facial expressions without rigid head motion. The database was constructed to contain an important quantity of dynamic natural expressions, both extreme and subtle, categorical and mixed. A crucial aspect of the analysis is that the captured expressions do not carry

any emotional label. The facial images will allow us to model the deformation of the face according to a scheme used in Section 3.1. The AAM procedure delivers a reduced set of parameters which represent the principal variation patterns detected on the face. Every facial expression can be projected onto this parameter space referred to as the appearance space (Figure 14 presents symbolic 3D representations of this space, although it may contain 15 to 20 dimensions). Note that this process is invertible: it is always possible to project a point of the appearance space back to a facial configuration, and thus synthesize the corresponding facial expression as a facial image.

5.1.2. A-Database. A reduced parameter space similar to the one described above can be constructed for the synthetic face, provided that a database of facial expressions for the virtual character is available (A-database). In this section we show how to identify a reduced set of facial configurations from the human database so that a coherent appearance space is constructed for the avatar (typically 25 to 30 expressions). The purpose of this avatar database creation scheme is that the appearance spaces of the human and the synthetic face have the same semantical meaning, and model the same information. It is then easy to construct a mathematical link between them (the ATM as illustrated on Figure 14).

The appearance space for the synthetic face is built through statistical modeling, similarly to the human appearance space (Section 5.1.1). For real faces, thousand of database samples can be produced with a video camera and a feature-tracking algorithm, whereas the elements of an equivalent synthetic database are manually-designed facial configurations, which are not easy to obtain. It is thus desirable to keep the number of required samples small.

Our idea for building the A-database, is to use the human database, and extract the expressions that have an important impact on the formation on the appearance space. Indeed, a lot of samples from the human database bring redundant information to the modeling process, and are therefore not essential in the A-database. Following this logic, we are able to reduce the set of necessary expression to a reasonable size.

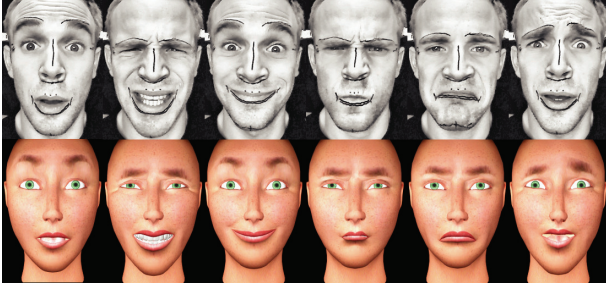


FIGURE 15: The first elements of the human expression located on the convex hull of the point cloud formed by all database elements (top). Avatar's expressions corresponding to each human expression (bottom).

Practically, We select the extreme elements of the database, meaning the elements presenting the maximal variations with respect to a neutral facial expression. In terms of parameter space, these elements are located on the convex hull of the point cloud formed by all database elements and are detected using [48]. These samples are responsible for shaping the meaningful variance of the database and thus encompass the major part of its richness. By manually reproducing these selected expressions on the face of the virtual character, we can build its very own appearance model according to the method presented in Section 3.1. Our studies have shown that 25–30 expressions are enough to train an efficient appearance model.

For the human database, we used more than 4000 elements. Using the convex hull procedure we have been able to identify 25–30 representatives for the reduced database (see Figure 15), with a small reconstruction error. Such a reduced database can be constructed for any synthetic character, and any human face based on the same extracted elements (see construction of the gamer's database in Section 6.2). Having to design several facial configurations manually on a synthetic character is a limitation of the method, yet it also can be seen as an advantage: our system does not rely on any particular facial control method (muscle systems, blendshapes, etc). Any scheme able to provide good facial configurations can be used. Our system can therefore easily be integrated in already-established workflows.

The database construction method creates a specific connection between the two databases, and thus the two appearance spaces. In the next sections, we will see how we benefit from it to animate the avatar based on the human motion data.

5.1.3. Appearance Transformation Matrix (ATM). The ideas developed in the previous section have lead to the construction of analogous appearance spaces for the human face and the synthetic face. Both spaces are connected, since the construction of the avatar appearance space is based on elements replicated from the human database. It follows that we have a correspondences between points in the human appearance space and points in the avatar space. We propose to use this sparse correspondence to construct an analytical

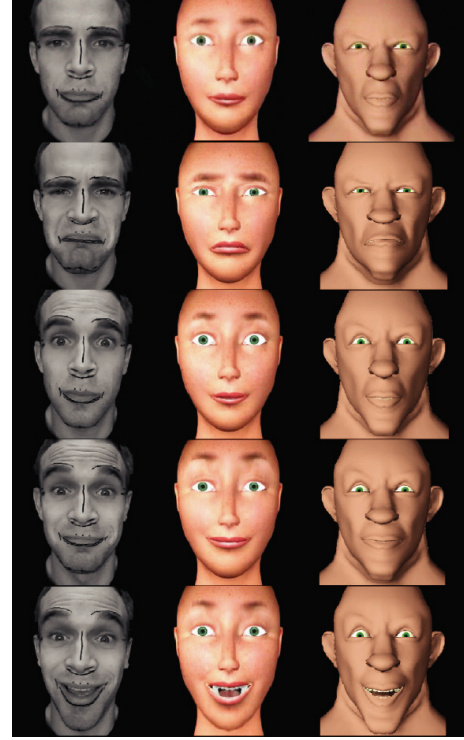


FIGURE 16: Examples of cloning of facial expressions. The expressions captured on the human face (left) are successfully transferred to the faces of avatars (middle and right). First row shows neutral faces.

link between both spaces. This link will then be used to transform human appearance parameters C_H into avatar appearance parameters C_A , and thus clone a human facial expression on the synthetic face.

It can be noted that the modeling scheme of AAM we use is linear equations (1), (2) and (3). Linear variations and combinations are thus preserved by the modeling steps, and we wish to maintain this linear chain in the retargeting process. Therefore, as in other approaches like [27], we applied a simple linear mapping on the parameters of the appearance spaces:

$$\begin{aligned}
 & \begin{vmatrix} C_{H(11)} & C_{H(12)} & \cdots & C_{H(1k)} \\ C_{H(21)} & C_{H(22)} & \cdots & C_{H(2k)} \\ \vdots & \vdots & \ddots & \vdots \\ C_{H(m1)} & C_{H(m2)} & \cdots & C_{H(mk)} \end{vmatrix} \\
 & = A_0 * \begin{vmatrix} C_{A(11)} & C_{A(12)} & \cdots & C_{A(1k)} \\ C_{A(21)} & C_{A(22)} & \cdots & C_{A(2k)} \\ \vdots & \vdots & \ddots & \vdots \\ C_{A(n1)} & C_{A(n2)} & \cdots & C_{A(nk)} \end{vmatrix} \quad (10)
 \end{aligned}$$

where m and n are the number of appearance parameters of human and synthetic appearance space respectively, while k is the number of expression stored in the database. Hence if

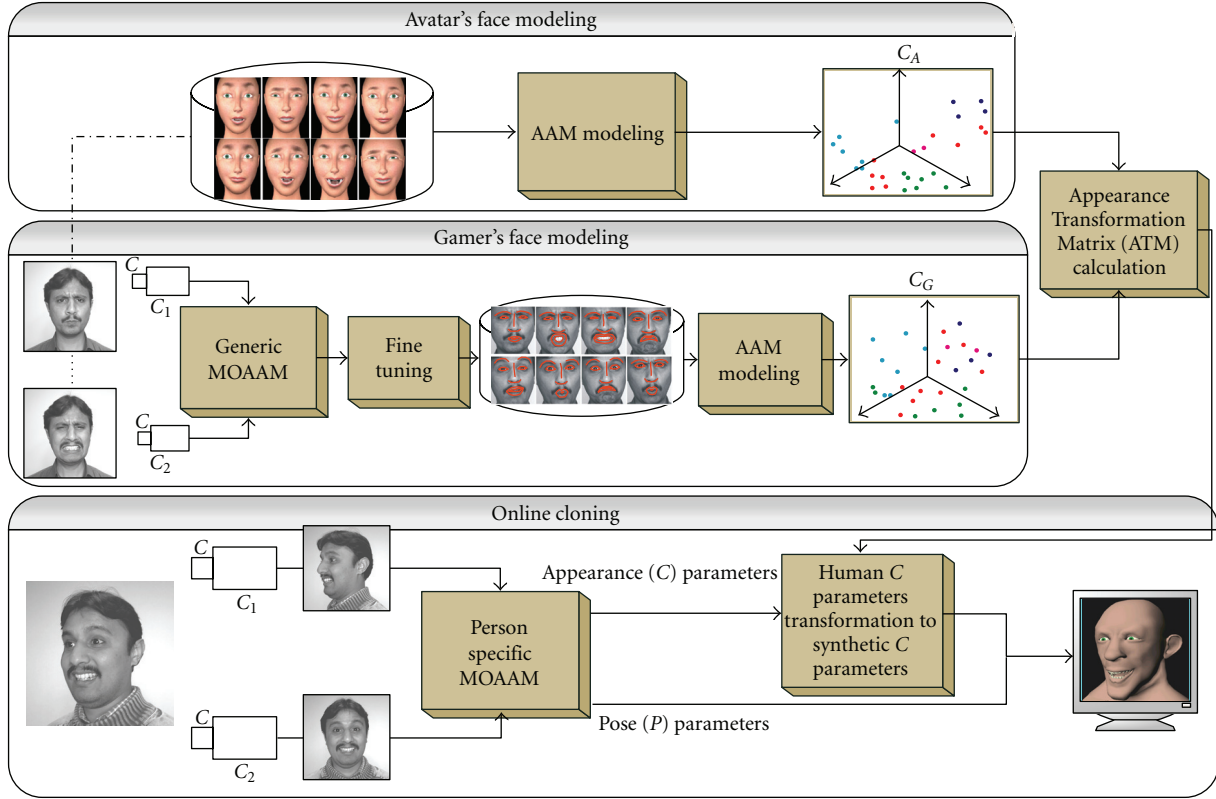


FIGURE 17: Block diagram of the interactive system.

TABLE 1: Experimental details. (Avatar1 and Avatar2 are shown in the middle and right columns of Figure 16 resp.).

	C-Parameters	Expression (k)
Human	$m = 24$	28
Avatar1	$n = 20$	28
Avatar2	$n = 18$	28

C_H is a $m \times k$ matrix and C_A is a $n \times k$ matrix, A_0 will be of $m \times n$.

The matrix A_0 is obtained through linear regression on the set of corresponding points. Depending on the dimensionality of the appearance spaces (usually 15 to 20), it can be profitable to turn to Principal Component Regression [49] to cope with a possible underdetermination of the regression problem. Retargeting results are illustrated by a few snapshots on Figure 16. Experimental details used are given in Table 1. Complete sequences of expression retargeting can also be found on the accompanying video.

6. Interactive System

Our proposition is a complete human machine interactive system for a game console. Figure 17 is a detailed description of our system. This time it is viewed from perspective of stages of the global system. System is composed of three stages.

6.1. Avatar's Face Modeling. In this section, we make use of procedure of Section 5.1.2 to obtain a database of simple and realistic facial expressions of an avatar called A-database. The visual aspect of the synthetic character is chosen by the user. Different classes of synthetic faces are available representing different ages, races, gender, physique and features and so forth. Once the class of the avatar is chosen, the required facial expressions, already stored in the system, are generated for this face (from the expressions identified in Section 5.1.2). Note that the system's user has the possibility to edit the suggested facial expression to personalize the look of its avatar by manually clicking and moving the vertices. Ultimately the A-database contains the expressions, on the user-chosen character, which are necessary to form the A-Database.

We can build the its appearance model according to the method presented in Section 3.1. This procedure delivers a reduced set of parameters which represent the principal variation patterns observed on the synthetic face (C_A). Manual marking of the landmark on the synthetic face is not needed as the synthetic face is already generated by the system and it contains the location of each vertex.

6.2. Gamer's Face Modeling. The procedure of training is very simple and unproblematic. The essence of this phase is to make the system learn the facial deformations of the gamer's face so it can replicate the localization of features, emotions and gestures on the synthetic face. The construction of the

TABLE 2: Processing time for Gamer's face modeling. (offline).

Processing block	Time
Generic MOAAM	90 milliseconds/frame
Fine tuning	30–40 sec/image
AAM modeling	5.68 sec (28 expressions)
Computation of ATM	24 milliseconds

Gamer's database is similar to the one of the avatar. The gamer has to mimic the expressions that have an important impact on the formation of the appearance space (identified in Section 5.1.2). In practice, the required facial expressions are displayed serially for the user to imitate. Facial images are captured by generic MOAAM, as explained in Section 4 to automatically localize the facial features. Since user is unknown to the system therefore generic MOAAM containing an AAM model based on M2VTS facial images database is used. Feature localized by MOAAM is displayed on the screen for the user to fine tune the location of each feature. Finally all the facial images of the gamer are generated, each corresponding to synthetic facial expression of the A-Database. By reproducing these selected facial expressions of the gamer, we can build its very own appearance model along with its reduced appearance parameters C_G according to the method presented in Section 3.1. With C_G and C_A (obtained in previous section) we can calculate ATM mathematically (see Section 5.1.3). This ATM is gamer dependent and can be used for cloning only for particular gamer who was involved in generating it in the first place. Time cost for this phase is tabulated in Table 2.

$$C_A = A_0 * C_G. \quad (11)$$

6.3. Online Cloning. From the previous two sections we obtained an ATM capable of transforming the appearance parameters from the gamer's appearance space to the avatar's appearance space. In online cloning, this transformation involves only a matrix multiplication of real-time gamer's appearance parameters C_G with A_0 to obtain avatar's appearance parameters C_A . This analytically simple framework enables real-time performances. The virtual illustration of a gamer is cloned in the form of an avatar synthesized by C_A and ultimately display on the screen as shown on Figure 17.

The appearance parameters of a gamer are acquired in real-time by our facial analysis system of multiple cameras. Tactical moves of the game causes the gamer to move a lot in different direction. Yet the retargeting scheme of Section 5.1 has been designed for stable heads. Employing multiple cameras resolved this problem. Two cameras placed at the extreme edges of the screen acquire real-time image of the gamer and at the same time his facial features and pose are analyzed by person specific MOAAM. Person specific MOAAM model is generated from the gamer database of the previous section and it contain all the *pose-free* facial variations of the gamer.

User's oriented face is analysed by MOAAM, to give its appearance and pose parameters. These appearance

TABLE 3: Processing time for online cloning.

Processing block	Time (msec/frame)
Person specific MOAAM	34
Transformation by ATM	0.015
Rendering (800 × 600) (2740 vertices)	30

parameters are pose-free and belongs to the frontal face of the user. These parameters are transformed by ATM in the synthetic face's parameter space and synthetic face is synthesized by them. After that pose parameters obtained previously by MOAAM analysis are used to adjust the orientation of the avatar being displayed on the screen. As shown in the cloning section of the Figure 17, appearance parameters undergoes transformation while pose parameter are directly reproduced on the avatar face to clone both the gamer's expressions and gestures. Time cost of each block, for a Pentium-IV 3.2 GHz platform, is tabulated in Table 3. The linearity of the AAM scheme allows the reproduction of both extreme and intermediate facial expressions and movements, with low computing requirements.

7. Conclusions

In this paper we proposed a solution to solve two bottlenecks of facial analysis and synthesis in an interactive system of human face cloning for nonexpert users of computer games. Facial emotions and pose of gamers cloned to bring their realistic behavior to virtual characters. Bottlenecks of analyzing the human face and synthesizing it in the form of an avatar are dealt with.

Large lateral movements of a gamer makes it impossible to analyze and track his face with single camera. To overcome this dilemma we exploit data from another camera and associate it with the one unable to analyze at the first place. Earlier the cost of a webcam and slow processor demotivated the possibility of managing excessive amount of data from multiple cameras. Currently with wide availability of inexpensive webcams the multiview system is as practical as single-view. To analyze the acquired multiview facial images we proposed multiobjective 2.5D AAM (MOAAM) optimized by Pareto-based NSGA-II. We have presented new results (Section 4.2) because of the problem of calibration and ground truth points in our previous work. Our approach of MOAAM is accurate, robust and capable of extracting the pose, features and gestures even with large lateral movements of a face.

As far as facial synthesis is concerned, cloning the human facial movements onto an avatar is not trivial due to their facial morphological differences. We proposed a new technique of calculating the mathematical semantic correspondence between the appearance parameters of the human and avatar (ATM matrix). We calculated this ATM for the gamer to be able to clone his emotions on the avatar in real-time. The interactive system we have presented is complete and easy to use. We have shown the results of facial

features and pose extraction and how we synthesize these facial details on an avatar by calculating the ATM with the gamer's help.

Although gamer's and avatar's database construction and its training is a long and tedious job. But it is supposed to be done once every time a new gamer is introduced. On the other hand our system is capable of performing online cloning of each frame in 64.015 milliseconds (i.e., 15 frames per second), as being nearly a real-time system. For the moment, this approach is limited to be used in an interactive system for the gamers, but it would be interesting to extend it for larger events, like conferences and meetings, with multiple cameras installed on different corners of the room and displayed on video projectors. Moreover it can be used efficiently in communication where the channel bandwidth is limited, since only the small amount of appearance and pose parameters are transmitted from the human face to the avatar for face synthesis.

Acknowledgments

Part of this work is financially supported by Higher Education Commission Pakistan. The authors would also like to thank the anonymous reviewers.

References

- [1] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proceedings of the European Conference on Computer Vision (ECCV '98)*, vol. 2, pp. 484–498, Freiburg, Germany, June 1998.
- [2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-II," in *Parallel Problem Solving from Nature VI Conference*, pp. 849–858, 2000.
- [3] A. Sattar and R. Seghier, "MVAAM (multi-view active appearance model) optimized by multi-objective genetic algorithm," in *Proceedings of the 8th IEEE International Conference on Automatic Face and Gesture Recognition (FG '08)*, 2008.
- [4] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for computer vision," Tech. Rep., Imaging Science and Biomedical Engineering, University of Manchester, 2004.
- [5] S. Ting, B. C. Lovell, and C. Shaokang, "Face recognition robust to head pose from one sample image," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 1, pp. 515–518, 2006.
- [6] X. Shenjun and A. Haizhou, "Face alignment under various poses and expressions," in *Proceedings of the 1st International Conference on Affective Computing and Intelligent Interaction (ACII '05)*, vol. 3784, p. 3784, Beijing, China, 2005.
- [7] L. Yanghua, L. Yang, T. Linmi, and X. Guangyou, "Multiview face alignment guided by several facial feature points," in *Proceedings of the 3rd International Conference on Image and Graphics (ICIG '04)*, pp. 238–241, 2004.
- [8] S. Yan and C. QianSheng, "Multi-view face alignment using direct appearance models," in *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '02)*, pp. 238–241, 2002.
- [9] M. Romero and A. F. Bobick, "Tracking head yaw by interpolation of template responses," in *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '04)*, vol. 5, p. 83, 2004.
- [10] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylors, "Coupled-view active appearance models," in *Proceedings of British Machine Vision Conference*, vol. 1, pp. 52–61, 2000.
- [11] F. Dornaika and J. Ahlberg, "Fast and reliable active appearance model search for 3d face tracking," in *Proceedings of Mirage INRIA Rocquencourt*, Paris, France, 2003.
- [12] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-time combined 2D+3D active appearance models," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 535–542, 2004.
- [13] J. Sung and D. Kim, "Extension of AAM with 3D shape model for facial shape tracking," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, vol. 5, pp. 3363–3366, 2004.
- [14] A. Sattar, Y. Aidarous, and R. Seghier, "GAGM-AAM: a genetic optimization with gaussian mixtures for active appearance models," in *Proceedings of the 15th International Conference on Image Processing (ICIP '08)*, pp. 3220–3223, 2008.
- [15] C. Hu, J. Xiao, I. Matthews, S. Baker, J. F. Cohn, and T. Kanade, "Fitting a single active appearance model simultaneously to multiple images," in *Proceedings of the British Machine Vision Conference*, 2004.
- [16] D. Kim and J. Sung, "A real-time face tracking using the stereo active appearance model," in *Proceedings of the International Conference on Image Processing (ICIP '06)*, pp. 2833–2836, 2006.
- [17] F. Dornaika and A. Sappa, "Improving appearance-based 3d face tracking using sparse stereo data," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP '06)*, 2006.
- [18] R. Yang and Z. Y. Zhang, "Model-based head pose tracking with stereovision," in *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '02)*, p. 255, 2002.
- [19] J. Tu, T. Huang, Y. Xiong, T. Rose, and F. K. H. Quek, "Calibrating head pose estimation in videos for meeting room event analysis," in *Proceedings of the International Conference on Image Processing (ICIP '06)*, vol. 5, pp. 3193–3196, 2006.
- [20] J. Chai, J. Xiao, and J. Hodgins, "Vision-based control of 3d facial animation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '023)*, pp. 193–206, 2003.
- [21] L. Williams, "Performance-driven facial animation," in *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '90)*, pp. 235–242, Dallas, Tex, USA, September 1990.
- [22] F. Pighin, R. Szeliski, and D. H. Salesin, "Resynthesizing facial animation through 3D model-based tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '99)*, vol. 1, pp. 143–150, 1999.
- [23] J.-Y. Noh and U. Neumann, "Expression cloning," in *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '81)*, pp. 21–28, 2001.
- [24] Z. Liu, Y. Shan, and Z. Zhang, "Expressive expression mapping with ratio images," in *Proceedings of the 19th Annual ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 271–276, 2001.
- [25] C. Kouadio, P. Poulin, and P. Lachapelle, "Real-time facial animation based upon a bank of 3d facial expressions," in *Proceedings of the Computer Animation*, pp. 128–136, 1998.

- [26] E. S. Chuang and C. Bregler, "Performance driven facial animation using blendshape interpolation," Tech. Rep. CS-TR-2002-02, Stanford University, 2002.
- [27] Y. Zhang, M. Luo, and S. Xu, "An efficient markerless method for resynthesizing facial animation on an anatomy-based model," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '07)*, pp. 971–974, 2007.
- [28] F. I. Parke, "Parameterized models for facial animation," *IEEE Computer Graphics and Applications*, vol. 2, no. 9, pp. 61–68, 1982.
- [29] M.-L. Viaud and H. Yahia, "Facial animation with wrinkles," in *Eurographics Workshop on Animation and Simulation*, 1992.
- [30] N. Magnenat-Thalmann, E. Primeau, and D. Thalmann, "Abstract muscle action procedures for human face animation," *The Visual Computer*, vol. 3, no. 5, pp. 290–297, 1988.
- [31] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann, "Simulation of facial muscle actions based on rational free form deformations," *Computer Graphics Forum*, vol. 11, no. 3, pp. 59–69, 1992.
- [32] K. Waters, "A muscle model for animation three-dimensional facial expression," in *Proceedings of the 3rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*, vol. 21, pp. 17–24, 1987.
- [33] E. Sifakis, I. Neverov, and R. Fedkiw, "Automatic determination of facial muscle activations from sparse motion capture marker data," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 417–425, 2005.
- [34] P. Ekman and W. V. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*, Consulting Psychologists Press, 1978.
- [35] A. Eleftheriadis, C. Herpel, G. Rajan, and L. Ward, "Mpeg-4 systems, text for iso/iec fcd 14496-1 systems," in *MPEG-4 SNHC*, 1998.
- [36] M. Byun and N. I. Badler, "Facemote: qualitative parametric modifiers for facial animations," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, vol. , pp. 65–71, 2002.
- [37] C. Curio, M. Breidt, M. Kleiner, Q. C. Vuong, M. A. Giese, and H. H. Bülthoff, "Semantic 3D motion retargeting for facial animation," in *Proceedings of the Symposium on Applied Perception in Graphics and Visualization (APGV '06)*, pp. 77–84, 2006.
- [38] E. S. Chuang and C. Bregler, "Mood swings: expressive speech animation," *ACM Transactions on Graphics*, vol. 24, no. 2, pp. 331–347, 2005.
- [39] I. Macedo, E. V. Brazil, and L. Velho, "Expression transfer between photographs through multilinear AAM's," in *Proceedings of the Brazilian Symposium of Computer Graphic and Image Processing (SIBGRAPI '06)*, pp. 239–246, 2006.
- [40] B.-J. Theobald, I. A. Matthews, J. F. Cohn, and S. M. Boker, "Real-time expression cloning using appearance models," in *Proceedings of the 9th International Conference on Multimodal Interfaces (ICMI '07)*, pp. 134–139, 2007.
- [41] L. Zalewski and S. Gong, "2D statistical models of facial expressions for realistic 3D avatar animation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 2, pp. 217–222, San Diego, Calif, USA, 2005.
- [42] D. Cosker, S. Roy, P. L. Rosin, and D. Marshall, "Re-mapping animation parameters between multiple types of facial model," *Lecture Notes in Computer Science*, vol. 4418, pp. 365–376, 2007.
- [43] M. B. Stegmann, *Active appearance models: theory, extensions and cases*, masterthesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2000.
- [44] C. Goodall, "Procrustes methods in the statistical analysis of shape," *Journal Royal Statistical Society*, vol. 53, pp. 285–339, 1991.
- [45] "Camera calibration toolbox," http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.
- [46] S. Pigeon, "M2VTS: multi modal verification for teleservices and security applications," 1996.
- [47] N. Stoiber, R. Seghier, and G. Breton, "Automatic design of a control interface for a synthetic face," in *Proceedings of the International Conference on Intelligent User Interfaces (IUI '09)*, 2009.
- [48] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.
- [49] J. O. Rawlings, *Applied Regression Analysis*, Wadsworth and Brooks Cole, 1988.

