

Advances in Artificial Intelligence

Machine Learning Paradigms for Modeling Spatial and Temporal Information in Multimedia Data Mining

Guest Editors: Djamel Bouchaffra, Abbes Amira, Ce Zhu, and Chu-Song Chen





**Machine Learning Paradigms for Modeling
Spatial and Temporal Information in
Multimedia Data Mining**

Advances in Artificial Intelligence

**Machine Learning Paradigms for Modeling
Spatial and Temporal Information in
Multimedia Data Mining**

Guest Editors: Djamel Bouchaffra, Abbas Amira, Ce Zhu,
and Chu-Song Chen



Copyright © 2010 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in volume 2010 of "Advances in Artificial Intelligence." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Advances in Artificial Intelligence

Editorial Board

Adel M. Alimi, Tunisia

Eduardo Alonso, UK

Aladdin Ayesh, UK

Bikramjit Banerjee, USA

Roman Bartak, Czech Republic

Daniel Berrar, UK

Cyrille Bertelle, France

Djamel Bouchaffra, USA

António D. P. Correia, Portugal

D. Girimonte, The Netherlands

David Glass, UK

Bernhard Graimann, Germany

Jun He, UK

Rattikorn Hewett, USA

Pascal Hitzler, USA

Christian Hölscher, UK

Jun Hong, UK

Fakhreddine Karray, Canada

Weiru Liu, UK

Bruce J. MacLennan, USA

Mark McCartney, UK

Gerard McKee, UK

Giorgio Metta, Italy

Ian Mitchell, UK

Richard Mitchell, UK

Iveta Mrazova, Czech Republic

Debajyoti Mukhopadhyay, India

Barry O'Sullivan, Ireland

Jeff Z. Pan, UK

Dave Patterson, UK

Martin Pelikan, USA

Guilin Qi, Germany

Alfons Schuster, UK

Alaa Fathy Sheta, Jordan

Shiliang Sun, China

Peter Tino, UK

Vincent Vidal, France

Jonathan Vincent, UK

Farouk Yalaoui, France

Filip Zelezny, Czech Republic

Contents

Machine Learning Paradigms for Modeling Spatial and Temporal Information in Multimedia Data Mining, Djamel Bouchaffra, Abbas Amira, Ce Zhu, and Chu-Song Chen
Volume 2010, Article ID 312350, 2 pages

Computing with Biologically Inspired Neural Oscillators: Application to Colour Image Segmentation, Ammar Belatreche, Liam Maguire, Martin McGinnity, Liam McDaid, and Arfan Ghani
Volume 2010, Article ID 405073, 21 pages

Evaluation of Data Quality and Drought Monitoring Capability of FY-3A MERSI Data, Daxiang Xiang, Liangming Liu, Qiao Wang, Na Yang, and Tao Han
Volume 2010, Article ID 124816, 10 pages

Multibandwidth Kernel-Based Object Tracking, Aras Dargazany, Ali Soleimani, and Alireza Ahmadyfard
Volume 2010, Article ID 175603, 15 pages

3D Medical Volume Segmentation Using Hybrid Multiresolution Statistical Approaches, Shadi AlZu'bi and Abbas Amira
Volume 2010, Article ID 520427, 15 pages

Unsupervised Topographic Learning for Spatiotemporal Data Mining, Guénaél Cabanes and Younès Bennani
Volume 2010, Article ID 832542, 12 pages

Editorial

Machine Learning Paradigms for Modeling Spatial and Temporal Information in Multimedia Data Mining

Djamel Bouchaffra,¹ Abbas Amira,² Ce Zhu,³ and Chu-Song Chen⁴

¹Grambling State University, Grambling, LA 71245, USA

²Brunel University, Uxbridge, Middlesex UB83PH, UK

³Nanyang Technological University, Singapore 639798

⁴Institute of Information Science, Taipei 115, Taiwan

Correspondence should be addressed to Djamel Bouchaffra, dbouchaffra@ieee.org

Received 31 December 2010; Accepted 31 December 2010

Copyright © 2010 Djamel Bouchaffra 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.

Multimedia data mining and knowledge discovery is a fast emerging interdisciplinary applied research area. There is tremendous potential for effective use of multimedia data mining (MDM) through intelligent analysis. Diverse application areas are increasingly relying on multimedia understanding systems. Advances in multimedia understanding are related directly to advances in signal processing, computer vision, machine learning, pattern recognition, multimedia databases, and smart sensors. The main mission of this special issue is to identify state-of-the-art machine learning paradigms that are particularly powerful and effective for modeling and combining temporal and spatial media cues such as audio, visual, and face information and for accomplishing tasks of multimedia data mining and knowledge discovery. These models should be able to bridge the gap between low-level audiovisual features which require signal processing and high-level semantics. A number of papers have been submitted to the special issue in the areas of imaging, artificial intelligence; and pattern recognition and five contributions have been selected covering state-of-the-art algorithms and advanced related topics.

The first contribution by D. Xiang et al. “*Evaluation of data quality and drought monitoring capability of FY-3A MERSI data*” describes some basic parameters and major technical indicators of the FY-3A, and evaluates data quality and drought monitoring capability of the *Medium-Resolution Imager* (MERSI) onboard the FY-3A.

The second contribution by A. Belatreche et al. “*Computing with biologically inspired neural oscillators: application to color image segmentation*” investigates the computing capabilities and potential applications of neural oscillators,

a biologically inspired neural model, to gray scale and color image segmentation, an important task in image understanding and object recognition. The major contribution of this paper is the ability to use neural oscillators as a learning scheme for solving real world engineering problems.

The third paper by A. Dargazany et al. entitled “*Multi-bandwidth Kernel-based object tracking*” explores new methods for object tracking using the mean shift (MS). A bandwidth-handling MS technique is deployed in which the tracker reach the global mode of the density function not requiring a specific starting point. It has been proven via experiments that the Gradual Multibandwidth Mean Shift tracking algorithm can converge faster than the conventional kernel-based object tracking (known as the mean shift).

The fourth contribution by S. Alzu’bi et al. entitled “*3D medical volume segmentation using hybrid multi-resolution statistical approaches*” studies new 3D volume segmentation using multiresolution statistical approaches based on discrete wavelet transform and hidden Markov models. This system commonly reduced the percentage error achieved using the traditional 2D segmentation techniques by several percent.

Furthermore, a contribution by G. Cabanes et al. entitled “*Unsupervised topographic learning for spatiotemporal data mining*” proposes a new unsupervised algorithm, suitable for the analysis of noisy spatiotemporal Radio Frequency Identification (RFID) data. The new unsupervised algorithm depicted in this article is an efficient data mining tool for behavioral studies based on RFID technology. It has the ability to discover and compare stable patterns in a RFID signal, and is appropriate for continuous learning.

Finally, we would like to thank all those who helped to make this special issue possible, especially the authors and the reviewers of the articles. Our thanks go to the Hindawi staff and personnel, the journal Manager in bringing about the issue and giving us the opportunity to edit this special issue.

Djamel Bouchaffra
Abbes Amira
Ce Zhu
Chu-Song Chen

Research Article

Computing with Biologically Inspired Neural Oscillators: Application to Colour Image Segmentation

Ammar Belatreche, Liam Maguire, Martin McGinnity, Liam McDaid, and Arfan Ghani

Intelligent Systems Research Centre, School of Computing and Intelligent Systems, Faculty of Computing Engineering, University of Ulster, Magee Campus, Northland Road, Northern Ireland, Derry BT48 7JL, UK

Correspondence should be addressed to Ammar Belatreche, a.belatreche@ulster.ac.uk

Received 2 December 2009; Accepted 26 February 2010

Academic Editor: Abbes Amira

Copyright © 2010 Ammar Belatreche 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.

This paper investigates the computing capabilities and potential applications of neural oscillators, a biologically inspired neural model, to grey scale and colour image segmentation, an important task in image understanding and object recognition. A proposed neural system that exploits the synergy between neural oscillators and Kohonen self-organising maps (SOMs) is presented. It consists of a two-dimensional grid of neural oscillators which are locally connected through excitatory connections and globally connected to a common inhibitor. Each neuron is mapped to a pixel of the input image and existing objects, represented by homogenous areas, are temporally segmented through synchronisation of the activity of neural oscillators that are mapped to pixels of the same object. Self-organising maps form the basis of a colour reduction system whose output is fed to a 2D grid of neural oscillators for temporal correlation-based object segmentation. Both chromatic and local spatial features are used. The system is simulated in Matlab and its demonstration on real world colour images shows promising results and the emergence of a new bioinspired approach for colour image segmentation. The paper concludes with a discussion of the performance of the proposed system and its comparison with traditional image segmentation approaches.

1. Introduction

Image segmentation is a crucial problem in machine vision, as there is no generic method that can be applied to all types of images and the choice of a particular method is rather problem specific. The segmentation process consists of partitioning an image into its homogenous regions whose pixels share similar features. It plays an important role in computer-aided diagnosis schemes, image understanding, and object recognition systems. Although the human brain performs the task of image segmentation efficiently and with apparent ease, it is still a major challenge for computer vision systems and remains an open research field which encompasses a variety of applications such as pattern recognition, medical diagnosis, remote sensing, robotics, content-based information retrieval, and search engines, to name but a few. Over the last few decades of research in machine vision, many techniques have been produced, usually based on pixel classification, edge detection, or region growing [1–3]; yet

modelling the segmentation output in a network of neurons remains a challenging task. There is a large body of research work on image segmentation in different application fields and the number of publications is still growing every year. It is beyond the scope of this work to provide a full survey of the various techniques developed to date in different application fields; the reader is referred to the following references for a comprehensive review of the plethora of algorithmic and machine learning techniques used for scene segmentation [4–7]. However, the main focus of this research is on using biologically plausible and biologically inspired techniques which mimic the way the animal and human brain represents and process sensory information.

This paper investigates the computing capabilities and potential applications of biologically plausible neural oscillators to unsupervised colour image segmentation. A recent hypothesis in neuroscience is that segmentation of different objects in a visual scene is based on the temporal correlation of neural activity [8–10]. Accordingly, a population of

neurons which fire in synchrony would signal attributes of the same object. Also, neurons with asynchronous activity would participate in the formation of different objects [11]. The paper presents a neural system based on the synergy between networks of neural oscillators and Kohonen self-organising maps. A network of locally excitatory and globally inhibitory neural oscillators is first applied to gray scale images then extended to colour images. The HSV (Hue, saturation, Value) colour space is used and a Kohonen self-organising map-based colour reduction method is developed. Both chromatic components and local spatial characteristics of image pixels are used. The proposed system is applied to segmentation of real world scenes and segmentation is achieved through synchronisation and desynchronisation of the activity of populations of neurons. The paper highlights the key benefits of the proposed approach and its main differences with existing approaches.

Section 2 presents a brief review of neural oscillators and discusses their biological relevance and the current related research. Section 3 introduces the mathematical models that underlie the dynamics of neural oscillators and presents simulations of both single and coupled neural oscillators and their phase plane analysis. Section 4 discusses the feature binding problem, the synchronisation of neural oscillators' activity, and its relevance to computer vision. Section 5 presents the application of networks of neural oscillators to segmentation of grey scale images. Section 6 describes the proposed colour image segmentation system. Finally, Sections 7 and 8 discuss the findings of this research and conclude the paper.

2. Biologically Inspired Neural Oscillators

The model of neural oscillators used in this work represents an instance of a general model called "relaxation oscillators" which represents a large class of dynamic systems that emerge from many physical systems (e.g., engineering, mechanical, and biological systems) [12, 13]. The first observation of relaxation oscillators dates back to 1926 when van der Pol [14] studied the characteristics of a triode circuit which show self-sustained oscillations. In his study, Pol realised that the observed oscillations were almost sinusoidal for a certain range of parameters and that abrupt changes were exhibited for a different range of parameters. The name of the model reflects the system time constants, also called relaxation times. A relaxation oscillator is characterised by a slow time scale needed to charge the capacitor and a fast time scale for its quick discharge. The period of oscillation is proportional to the relaxation time needed to charge the capacitor, hence the name of the model.

Relaxation oscillators can be observed in a variety of biological phenomena such as heartbeat and neural activity. Some scientists, such as the physiologist Hill [15], went even further by stating that all physiological periodic phenomena are governed by this type of oscillations, that is, relaxation oscillations. However, despite the existence of such models in a number of domains, the real motivation of exploring them in this work comes from their relevance to neurobiology;

hence the focus of this study is oriented towards models that represent neural activity in the brain and the investigation of emerging behaviours from networks of such neural models (oscillators) and their potential application in solving real world engineering problems.

It has been recognised that relaxation oscillators are similar to their biological counterparts and were therefore used to model biological phenomena. They were used in 1928 by van der Pol and van der Mark to describe the heartbeat and to present an electrical model of the heart [16]. Using a system of differential equations based on four dynamic variables, Hodgkin and Huxley have mathematically described the dynamics of the neuron membrane potential, ionic exchange, and electric transmission of impulses along nerves [17]. Their famous system for which they obtained the Noble prize in "Physiology or Medicine" in 1963 (nobelprize.org) was later reduced to a system of two dynamic variables called Fitzhugh-Nagumo model [18, 19] which represents a relaxation oscillator. A relaxation oscillator description of the burst-generating mechanism in the cardiac ganglion cells of the lobster was later developed by Mayeri [20]. Oscillations in cortical neurons were also modelled by Wilson-Cowan polynomial equations which model interactions between excitatory and inhibitory neurons [21–24]. Their model consists of a two-variable system of differential equations with a number of parameters which offer a variety of dynamics when carefully adjusted [25]. Another model was presented by Morris and Lecar and consists of a two-variable relaxation oscillator which describe the voltage oscillation in the giant muscle fibre [26].

Since coherent oscillations were discovered in the visual cortex and other areas of the brain [27–29] and due to the neurobiological relevance of relaxation oscillators, they have attracted more research work where they have been studied as mathematical models representing the behaviour of neurons [30–36]. These research findings concluded that the observed oscillations are stimulus-dependent as they are triggered by appropriate sensory stimulus, that synchronisation of these oscillations emerges when the sensory stimuli appear to belong to a coherent object, and that no such synchrony is observed otherwise, that is, in the case where the stimuli are not connected by similarities or a common object. These findings are consistent with the principles of the theory of temporal correlation [37, 38] which explains how the perception of a coherent object is brought about by the brain through temporal correlation of the firing activity of various neurons which detect the features of the perceived object. This is also related to the binding problem which is explained later in Section 4.

Synchronisation of neural activity was observed in different areas of the brain [39, 40]. It has been observed in mammals, amphibians, and insects [29, 41, 42]. It has been measured in the monkey motor cortex [43] and across different hemispheres of the brain in the visual cortices of cats [44]. One of the main concerns of this research work is to understand how synchrony is brought about in locally interacting groups of neural oscillators and how the emerging-synchronised activity of populations of neural oscillators can be applied to image perception,

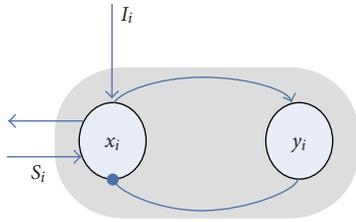


FIGURE 1: A single neural oscillator which consists of two coupled units x (excitatory) and y (inhibitory). Unit x sends excitation (indicated by a triangle) to unit y ; unit y sends inhibition (indicated by a small circle) to unit x . I_i and S_i represent an external input and a possible coupling with other neural oscillators, respectively.

particularly the important problem of image segmentation which is a crucial task in image understanding and object recognition. This motivation is driven by the fact that the exhibition of synchronised neural activity in a wide range of brain areas and by a diversity of organisms indicates that neural synchronisation could be fundamental to information processing.

3. Modelling Dynamics of Neural Oscillators

3.1. Dynamics of Single Neural Oscillators. The neural oscillator model used in this work is similar to the relaxation oscillator defined by Terman and Wang [36] which is based on the neural oscillator model developed by Morris and Lecar [26]. However the chosen model is considerably simpler and has been shown to achieve synchrony more rapidly than other neural oscillators models [45]. According to this model, a neural oscillator consists of a feedback loop between two units, one excitatory (called x_i) and the other is inhibitory (called y_i) as illustrated in Figure 1.

The dynamics of the temporal activity of the coupled units (x_i and y_i) are governed by the following system of first order differential equations:

$$\begin{aligned} \dot{x}_i &= 3x_i - x_i^3 + 2 - y_i + \rho + I_i + S_i, \\ \dot{y}_i &= \varepsilon \left[\lambda \left(1 + \tanh\left(\frac{x_i}{\beta}\right) \right) - y_i \right], \end{aligned} \quad (1)$$

where I_i represents an external input (e.g., pixel features) and S_i defines the total coupling from other neurons (in the case of a single oscillator, the coupling term does not exist, i.e., $S_i = 0$). The parameter ρ represents the amplitude of a Gaussian noise added to the total oscillator input in order to test its robustness to noise, and also to contribute to the desynchronisation process when a neural oscillator is coupled with other neurons. The parameter ε is a small positive number whose value affects the time scales (which characterise relaxation oscillators) of the activities of units x and y . The parameter β controls the steepness of the y -nullcline which will be explained below when the dynamics of the neural oscillator are analysed in the phase plane. The parameter λ controls the amount of the time spent on either phase (a larger value leads to a shorter time spent on the

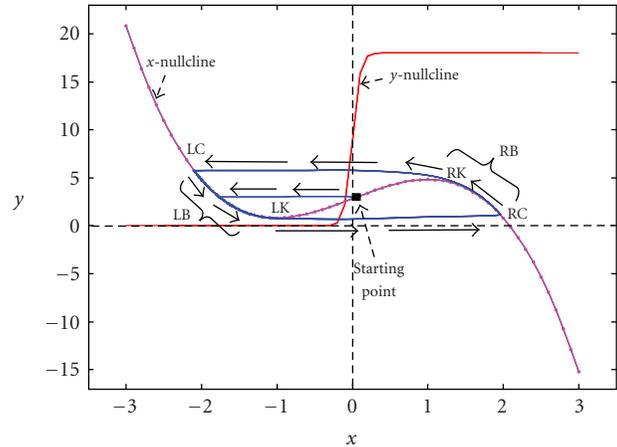


FIGURE 2: x - and y -nullclines with a periodic solution of the system in (1) (limit cycle) represented in the phase plane. This oscillator trajectory jumps periodically between the left and the right branches of the limit cycle. LC, LB, and LK indicate the upper left corner, left branch, and left knee, respectively. RC, RB, and RK indicates, lower right corner, right branch, and right knee, respectively. The starting point for this simulation is set to $(x_i, y_i) = (0, 3)$ and the values of other parameters are $\lambda = 9$, $\beta = 0.1$, $\varepsilon = 0.02$, $\rho = 0.02$, and $I = +0.8$. (Note that the parameter values are based on the range of values found in literature.)

active phase). In order to better understand the dynamics of a neural oscillator defined by the system in (1), the phase plane analysis approach is used where the oscillator nullclines, limit cycles, and motion directions are examined. Phase plane analysis is a graphical method which helps understand the behaviour of a dynamic system over time. The nullclines and limit cycle of the oscillator i defined in (1) are examined first. These figures were generated using Matlab implementation of the dynamics described above. The 4th-order Runge-Kutta method was used to numerically solve the system of first-order differential equations in (1).

The x -nullcline is represented by the curve along which the derivative of the unit x_i function is nil (i.e., $\dot{x}_i = 0$). It is given by the function $y_i = 3x_i - x_i^3 + 2 + \rho + I_i + S_i$ and results in a cubic curve that is characterised by a left branch denoted by LB (which extends from the local minimum to $-\infty$) and a right branch denoted by RB (which extends from the local maximum to $+\infty$) as illustrated in Figure 2. This cubic curve has two crucial values (related to the dynamics of a neural oscillator) which are defined by the function values (i.e., the y -values) at the two local extrema and referred to by left knee (LK) and right knee (RK) (see Figure 2). On the other hand, the y -nullcline is determined by the curve along which the derivative of the unit y function is nil (i.e., $\dot{y}_i = 0$). It is given by the function $y_i = \lambda[1 + \tanh(x_i/\beta)]$, a hyperbolic tangent function which results in a sigmoidal curve whose steepness is controlled by the parameter β such that a big value of β makes the sigmoidal curve close to a step function. Based on these nullclines, the dynamics of a neural oscillator depend on the external input. In the presence of a positive input (i.e., the neuron is stimulated), the neuron dynamics converge

to a periodic solution, called limit cycle, and as shown in Figure 3. However, the neurons dynamics converge to stable point (i.e., no oscillation is produced) in the presence of a negative input (i.e., the neuron is unstimulated) as shown in Figures 4 and 5.

3.2. Dynamics of Coupled Neurons. In this section, the behaviour of two coupled neural oscillators is examined with particular focus on the emerging synchronisation and desynchronisation of their temporal activities which depends, amongst other parameters, on the coupling strength linking them together. The same model equations in the previous section are used for each neural oscillator with the addition, this time, of the coupling term (which was discarded in the case of a single oscillator). The connection between two neural oscillators is illustrated in Figure 6 which clearly shows that it is the unit x of the neural oscillator that sends (receives) output (input) to (from) another oscillator and that the role of unit y is only “internal” and consists of sending an inhibiting signal to the unit x of the same oscillator as explained in the previous section. As a result, the unit y has no interaction with other units of other oscillators.

The mathematical equations defining Oscillator i and Oscillator j are given, respectively, by (2) and (3) as follows:

$$\begin{aligned} \dot{x}_i &= 3x_i - x_i^3 + 2 - y_i + \rho + I_i + S_i(x_j), \\ \dot{y}_i &= \varepsilon \left[\lambda \left(1 + \tanh\left(\frac{x_i}{\beta}\right) \right) - y_i \right], \end{aligned} \quad (2)$$

$$\begin{aligned} \dot{x}_j &= 3x_j - x_j^3 + 2 - y_j + \rho + I_j + S_j(x_i), \\ \dot{y}_j &= \varepsilon \left[\lambda \left(1 + \tanh\left(\frac{x_j}{\beta}\right) \right) - y_j \right], \end{aligned} \quad (3)$$

where S_i represents the coupling term received by Oscillator i from Oscillator j , and S_j represents the coupling term received by Oscillator j from Oscillator i . The general form of this coupling term (which applies to the case where a neural oscillator is connected to several oscillators) is given by (4):

$$S_i = \sum_{k \in N_i} W_{ik} H(x_k - \theta_x), \quad (4)$$

where W_{ik} are positive synaptic weights (coupling strength) connecting neuron k and i , N_i represents the neighbourhood of neuron i , H is a Heaviside function, and θ_x is a threshold that is applied to the received input from neighbouring neurons.

From (4), it can be seen that the application of the Heaviside function on the received input allows an oscillator to receive its neighbour input only if the latter is above a certain threshold θ_x . Other transform functions can also be used (instead of the Heaviside function) such as a sigmoidal function. Only positive weights are assumed for the coupling term which mimics the excitatory synapses. For the applications which will be discussed in the following sections, neural oscillators will be connected locally in an excitatory way while a global common inhibitor is used to allow competition between groups of neurons.

The x and y -nullclines of both neural oscillators and the convergence of their trajectories to a limit cycle are represented in the phase plane as illustrated in Figure 7. The threshold is represented by a vertical line with a circle on the x -axis. Three nullclines are obtained for this simulation as the same y -nullcline is common to both oscillators; however two different x -nullclines are obtained because the change in the parameter values only affects the equations of unit x while the unit y equations remain identical (this change is caused by the use of different initial parameters and also during the continuous interaction defined by the coupling term). The upper cubic is referred to as the “excited x -nullcline/cubic” because it represents the excited neural oscillator (i.e., the neuron whose coupling term S_i is above the threshold). Likewise the lower cubic is referred to as the “unexcited cubic” as it represents the neural oscillator without excitation (i.e., with a nil coupling term $S_i = 0$).

Based on the coupling strength, the emerging behaviour is either a synchronisation or a desynchronisation of the neural oscillators’ temporal activities. Increasing the coupling strength leads to an elevation of the x -nullcline (the cubic); on the other hand a decrease in the coupling strength will shift the cubic downwards. The example shown in Figure 8 illustrates the dynamics of strongly coupled oscillators which led to synchronised activities (and to the neural oscillators’ trajectories converging to a limit cycle as shown in Figure 7).

The example shown in Figure 9 illustrates the dynamics of weakly coupled oscillators which led to desynchronised activities.

The interactions between coupled oscillators, which result in either a synchronisation or a desynchronisation of their temporal activities, will form the basis of the following sections which describe the application of networks of neural oscillators to grey scale and colour image segmentation developed in this work. However, before discussing the developed system in Sections 5 and 6, it is important to introduce the problem of feature binding and highlight its relation to the temporal synchronisation in neural oscillators and its relevance to object segmentation and image analysis tasks. The following two sections will briefly address these issues.

4. Relation to Digital Image Segmentation

There are two main competing hypotheses to explain how the brain binds together the distributed features of a perceived object (i.e., the binding problem [46–52]). The first hypothesis suggests that information about a stimulus is conveyed by the average firing rate of a single neuron and that neurons at higher brain areas become more selective [53–56] so that a single neuron may represent each single object (the grandmother-cell representation). Consequently, multiple objects in a visual scene would be represented by the coactivation of multiple units at some level of the nervous system. This hypothesis faces, however, major theoretical and neurobiological criticism which is rather in favour of another hypothesis which postulates that it is the temporal correlation of the firing activities of distributed neurons

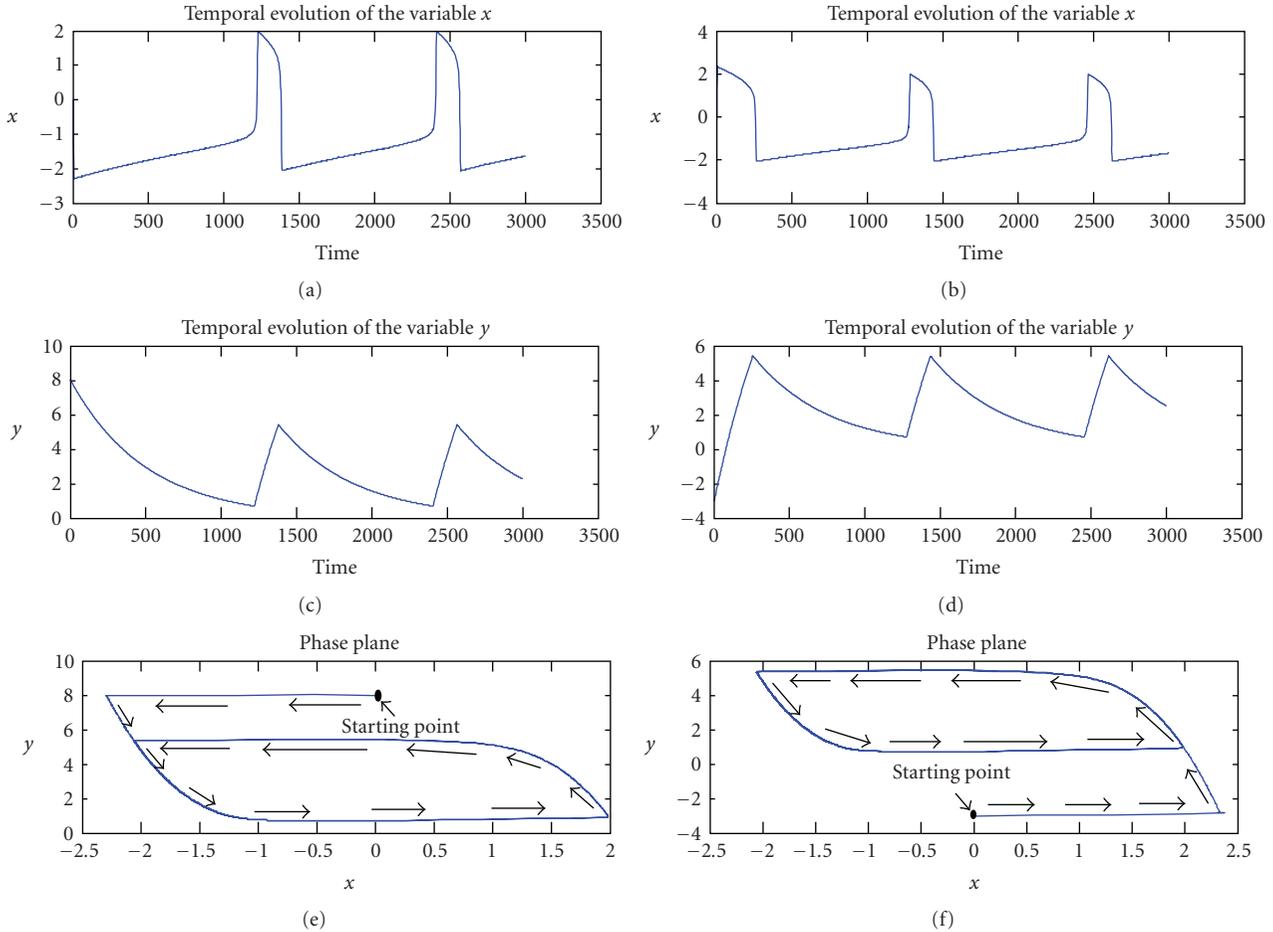


FIGURE 3: Dynamics of a single neural oscillator and its convergence to a periodic limit cycle. ((a), (c), and (e)) Temporal activity of the excitatory unit x which represent the output of the neural oscillator (a). Temporal activity of the inhibitory unit y (c). Trajectory of the neural oscillator with a starting point $(x, y) = (0, 8)$ (e). ((b), (d), and (f)) Temporal activity of the excitatory unit x (b). Temporal activity of the inhibitory unit y (d). Trajectory of the neural oscillator with a starting point $(x, y) = (0, 8)$ (f). It can be seen how the trajectory of the neural oscillator converges to the same stable periodic limit cycle. The same parameters in Figure 2 are used for this simulation.

that is used for binding the features of an object. This means that an object is coded by a population of many neurons whose firing activity is synchronised. According to this hypothesis, different objects can be encoded by different populations of neurons each of which has different firing times. Evidence of this approach has been established through several neurophysiological experiments (a review of these experiments can be found in [49, 57]).

Based on the second hypothesis and its theoretical and experimental supporting evidence, the work in this paper attempts to understand and exploit the emerging synchronisation in networks of locally connected neural oscillators and apply such self-organising behaviour to colour image segmentation. Although the focus of this work is on visual scenes (images) only, the idea of temporal correlation-based feature binding can be exploited in the processing of other sensory modalities such as speech signals. The image features (pixel attributes) are distributed over a network of neural oscillators where each feature is represented by a single neural oscillator. Therefore, the synchronisation of

a group of oscillators encodes the object they represent. On the other hand, the desynchronisation of other neural oscillators indicates that they do not belong to the same object and might rather represent other objects of the scene. Figure 10 illustrates the idea of temporal scene segmentation where existing objects are segmented and emerge through time as the temporal activity of its corresponding neural oscillators synchronise with each other and desynchronise with other neural oscillators which represent other objects.

We referred to the features of an image as “pixel attributes” instead of pixel values because they depend on the type of the processed image. That is, in a black and white (binary) or grey scale image the pixel values could be directly used as attributes/features, although the use of other features based on the values of neighbouring pixels is also possible and sometimes might be unavoidable (such as in texture images). In a colour image, on the other hand, the situation is different and the colour of a pixel cannot be directly represented by a single oscillator as it is

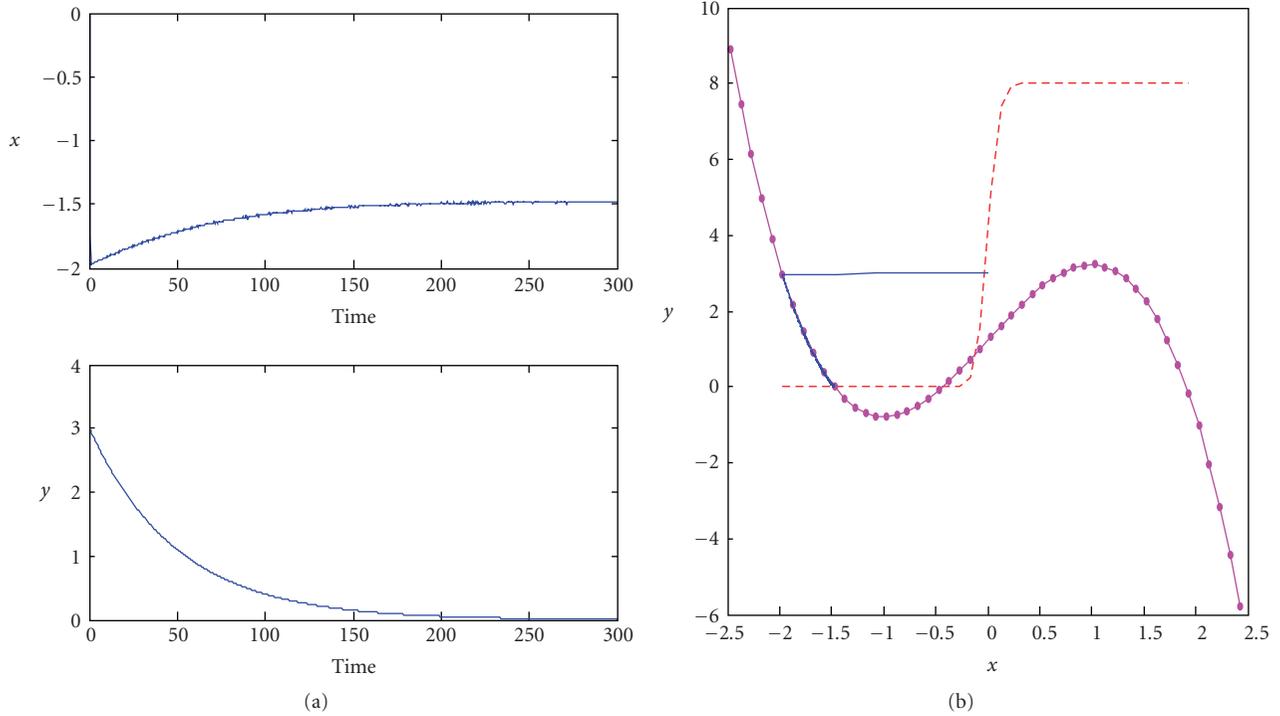


FIGURE 4: Activity of an unstimulated neural oscillator. (a) Temporal activity of units x and y with initial conditions set to $(x, y) = (0, 3)$. (b) Convergence of the oscillator trajectory to a fixed point (instead of a periodic limit cycle). The parameters used in this simulation are as follows: ($\lambda = 4$; $\beta = 0.1$; $\varepsilon = 0.02$; $\rho = 0.02$; $I = -0.8$).

encoded by a triplet of integer values. Therefore the feature value should be determined by some mechanism before being eventually fed to a neural oscillator. In this work we present the self-organising maps of Kohonen (SOM) as a feature extraction phase where the pixel values (triplets) of the original colour image are mapped to a new reduced space of colours where each pixel is assigned one value instead of a triplet. The creation of a new space and the mapping of original colours are based on the self-organising characteristic of Kohonen maps. The implementation details of this process are presented in Section 6. The following section discusses the application of neural oscillators to image segmentation, starting with the segmentation of grey scale images.

5. Application of Neural Oscillators to Grey Scale Image Segmentation

It is hypothesised in neuroscience that segmentation of different objects in a visual scene is based on the temporal correlation of neural activity. Accordingly, a population of neurons which fire in synchrony, or in a highly correlated way, would signal attributes of the same object. Also, neurons with asynchronous activity would participate in the formation of different objects. Based on these principles and the dynamics of neural oscillators introduced in previous sections, this work uses a network (grid) of neural oscillators that are locally connected through excitatory synapses and

globally inhibited by a common inhibitor as defined in [36]. Such a network is first applied to grey scale images where the intensities of pixels are directly assigned to neural oscillators then extended to colour images.

5.1. Network Architecture. The network consists of a grid (a 2D array) of neural oscillators which are locally connected through excitatory synapses represented by positive weights and a global neuron which receives excitation from each neuron and sends inhibition to all neurons in the grid (i.e., negatively connected to all neurons). Each neuron input is assigned a feature of the image pixel as illustrated in Figure 11. The local connections between each two neurons are reciprocal and each neuron is connected with its nearest four neighbouring neurons as shown in Figure 12. While a 4-connectedness is shown in this figure, it is still possible to use alternative forms such as 8-connectedness where each neuron is reciprocally connected to its 8 nearest neighbours. The motivation behind local excitatory connections is that, besides its consistence with lateral synaptic connections in several areas of the brain, it is hoped that such connections will contribute to object segmentation by inducing synchronised activities amongst neurons which belong to the same region.

The building block of the network is based on the model described in (1) and (2) with a slight modification of the external input term and the integration of the global inhibition in the coupling term S_i . The resulting equations

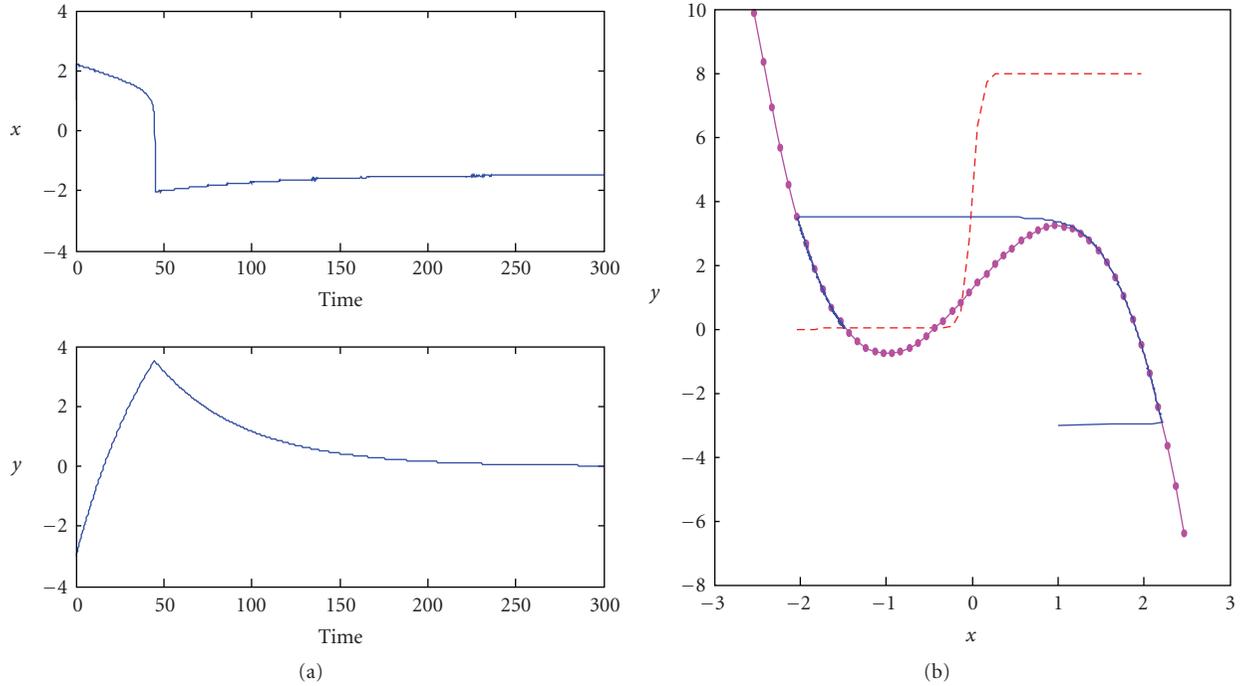


FIGURE 5: Another example of the same unstimulated neural oscillator in Figure 2 with a different starting point $(x, y) = (1, -3)$.

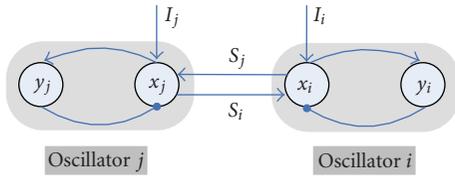


FIGURE 6: Example of two coupled neural oscillators. Besides the external input received and fed to each neural oscillator, both oscillators send/receive an output/input to/from each other.

are given by

$$\begin{aligned} \dot{x}_i &= 3x_i - x_i^3 + 2 - y_i + I_i H(p_i - \theta) + S_i + \rho, \\ \dot{y}_i &= \varepsilon \left(\gamma \left(1 + \tanh\left(\frac{x_i}{\beta}\right) \right) - y_i \right). \end{aligned} \quad (5)$$

The coupling from other neighbouring neurons is represented by S_i and defined by

$$S_i = \sum_{k \in Ni} W_{ik} H(x_k - \theta_x) - W_z z H(z - \theta_z). \quad (6)$$

Similarly to the coupling term defined in Section 3 in relation to coupled oscillators, excitatory connections (W_{ik}) are used and contribute to the propagation of the neuron activity to its neighbours. Also, an oscillator can receive its neighbour input only if the latter is above a certain threshold θ_x (set between LK and RC). The calculation of the connection weights of a neural oscillator is based on the value of its input (pixel feature) and those of its neighbours such as oscillators

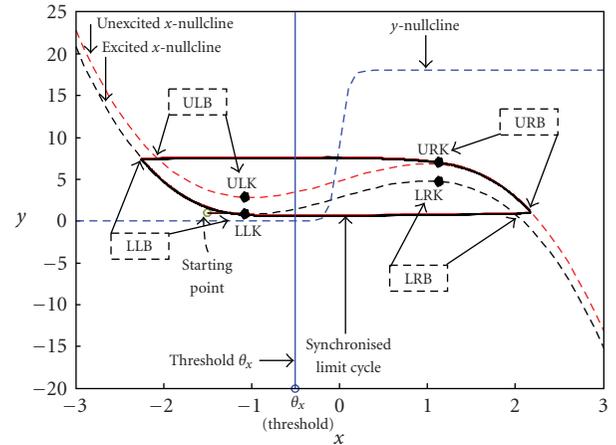


FIGURE 7: Nullclines of two-coupled oscillators and the convergence of their temporal activity to a synchronised limit cycle which is represented by a thick solid line. The x -nullclines (cubic curves) of the system in (2) and (3) are represented by a dashed line and the y -nullcline (sigmoid curve) is represented by a dotted line. The excitation threshold (θ_x) is set between the left and the right branches of the limit cycle (vertical line with a circle end on the x axis). ULB (upper left branch), URB (upper right branch), LLB (lower left branch), LRB (lower right branch), ULK (upper left knee), LLK (lower left knee), URK (upper right knee), and LRK (lower right knee). The parameters used for this simulation are $W_{ij} = W_{ji} = 3.0, \theta_x = -0.5, \lambda = 9, \beta = 0.1; \varepsilon = 0.04; \rho = 0.02; I = +0.8$.

with similar pixel values are assigned strong weights while other oscillators belonging to different regions are given weaker connections. There is a new term added in this case

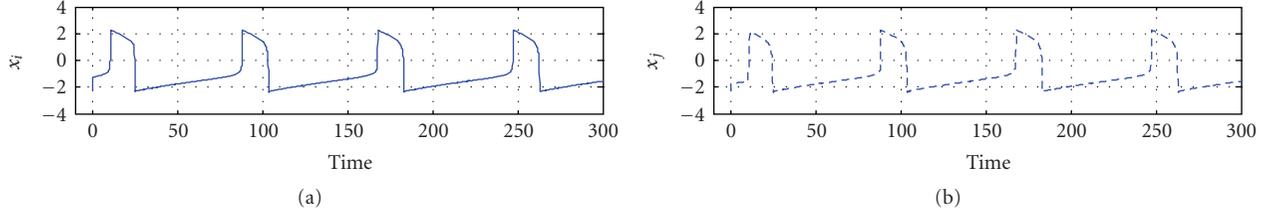


FIGURE 8: Synchronisation in a pair of oscillators. When the coupling between two neural oscillators is strong enough, a synchronisation between their temporal activity is obtained. Same parameters of Figure 7 are used with a starting point for oscillator i at $(x_i, y_i) = (-2.3, 1)$ and $(x_j, y_j) = (-2.3, 3)$ for oscillator j .

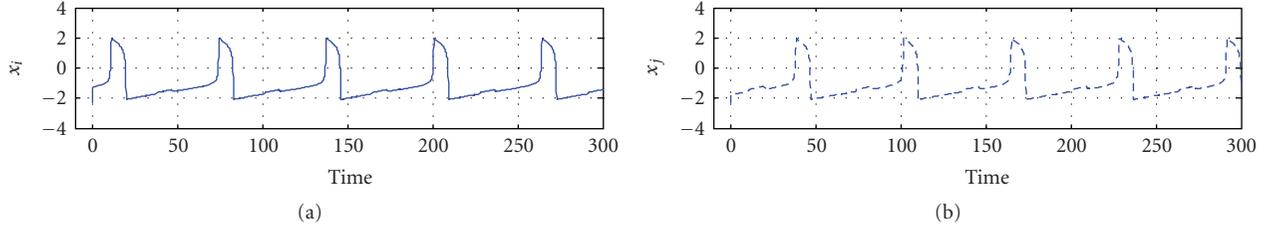


FIGURE 9: Desynchronisation in a pair of oscillators. A weak coupling ($W_{ij} = W_{ji} = 0.5$) between two neural oscillators is applied. Other parameters of the two neurons remain similar to those used in Figure 8.

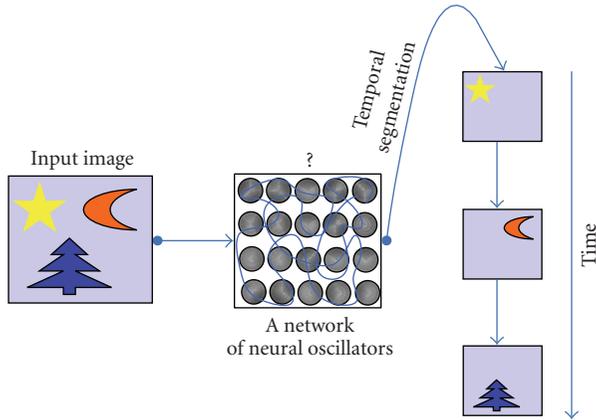


FIGURE 10: A colour image and its temporal segmentation using a network of neural oscillators. Neural oscillators which belong to the same object get their temporal activities synchronised with each other and desynchronised with other neurons from other different objects. This synchronisation and desynchronisation signal the detection of an object.

to count for the global inhibitor whose inhibition weight is denoted by W_z . The activity of the global inhibitor is represented by the dynamic variable z defined as follows:

$$\dot{z} = \phi(\sigma - z), \quad (7)$$

where $\sigma = 1$ if at least one oscillator is on the active phase, and $\sigma = 0$ otherwise. When an oscillator is triggered, $z \rightarrow 1$, and ϕ is a parameter which determines the rate at which the inhibitor reacts to the stimulation from an active oscillator. According to (6), the inhibitor activity is considered only when its value z is above a certain threshold θ_z (due to the

Heaviside function), in which case the term W_z is subtracted from the total excitation from neighbouring neurons. The global inhibitor, which receives excitatory input from all the neurons and sends back inhibitory outputs to all neurons, contributes to the desynchronisation of other oscillators which does not belong to the object being segmented. It cannot affect synchronised oscillators as the sum of the inputs from synchronised neighbours is greater than W_z .

In addition, the function p_i , called the lateral potential, is introduced in (5) in order to distinguish between small sparse noisy fragments and coherent regions so that oscillations of noisy fragments are removed. Major coherent region contains at least one oscillator (called leader) with large enough lateral excitation ($>\theta_p$) from its neighbours, but such a leader oscillator does not exist in a noisy fragmented region. Therefore the function p_i determines whether or not an oscillator is a leader and also plays a role in “ignoring” the noise in the input image. It is given by (8):

$$\dot{p}_i = \lambda(1 - p_i)H\left[\sum_{k \in N_i} W_{ik}H(x_k - \theta_x) - \theta_p\right] - \mu p_i, \quad (8)$$

where λ is a constant. If the weighted sum of active neighbours (each of which should obviously exceed θ_x to make the inner Heaviside evaluate to 1) is above a certain threshold θ_p , then the outer Heaviside is activated (becomes 1) and p_i approaches one; otherwise it relaxes to zero on a time scale determined by a small value μ . Thus, only oscillators which are surrounded by a large number of active oscillators can maintain their p_i high.

5.2. Numerical Simulations. In order to illustrate the separation of existing objects within a grey scale image through the synchronisation of neural oscillators belonging to the

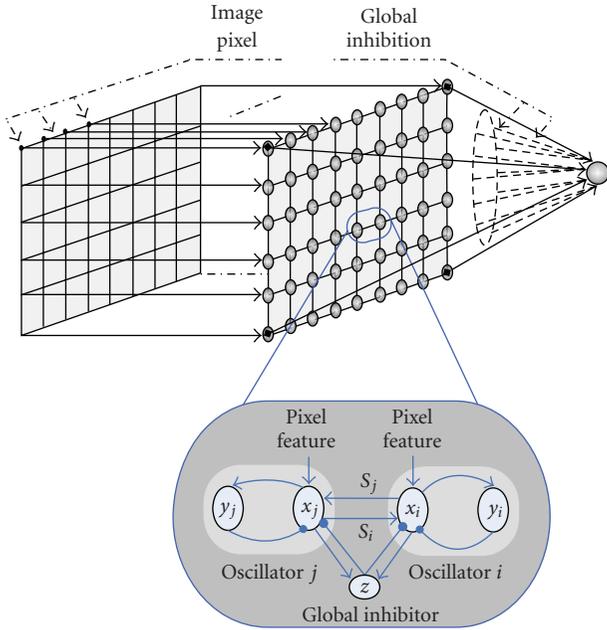


FIGURE 11: A grid (network) of competitive self-organising locally excitatory globally inhibitory neural oscillators. Each neural oscillator is connected to its four neighbours through excitatory synapses and receives inhibition from a global inhibitor. The latter helps create competition between neurons that belong to a single object (synchronised) and other neurons which belong to different objects (desynchronised). Only the coupling between oscillators x_i and x_j has been highlighted; the connections between each neuron and other neighbours are not shown for simplicity.

same object and desynchronisation of others belonging to different objects, tests on a small synthetic image and a larger real scene are presented below. While a discrete integration is used for the first test, however, for the second test, due to the computation intensive nature of discrete integration of a big number of oscillators, an algorithmic approach which captures the essence of the neural oscillators dynamics is used.

5.2.1. *Step-by-Step Equation Integration for Small Scale Images.* A network of 10-by-10 neural oscillators, based on the models described above and the behaviour of the grid architecture explained above, was numerically simulated using Matlab. The differential equations were first integrated using the Runge-Kutta 4th order to ensure a better accuracy. However the forward Euler method was also used and the same results were obtained. A 10-by-10 synthetic grey scale image, shown in Figure 13, is fed to a network of 100 neural oscillators. It consists of four separate rectangles, laid upon a black background, each of which has a different size and a different gray scale. Each pixel value is fed as an external input to the corresponding neural oscillator in the grid. The objects are labelled Obj 1, Obj 2, Obj 3, and Obj 4.

Figure 14 shows the temporal activity of all neural oscillators in the grid. Only the first 1800 integration steps are shown; however the oscillations carry on indefinitely unless

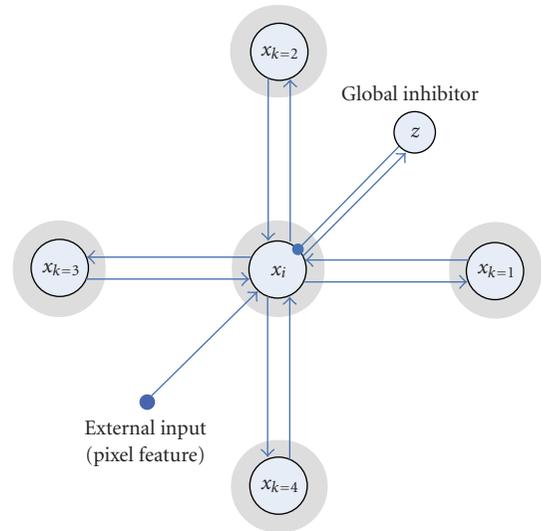


FIGURE 12: Connections between each neural oscillator, its four neighbours (x_1 , x_2 , x_3 , and x_4) and the global inhibitor (z). Every neural oscillator in a network is connected to the global inhibitor (these connections are not shown here for simplicity). Only the unit x of the oscillator is shown.

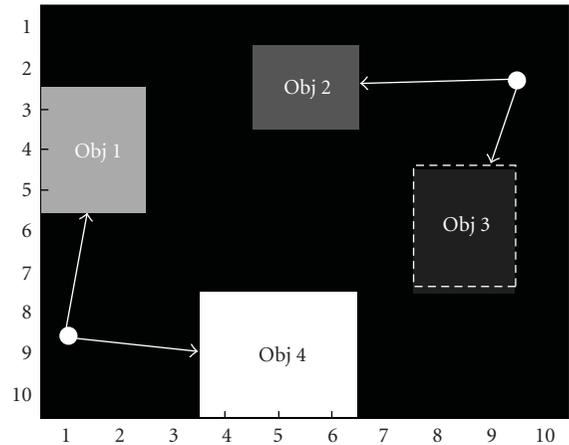


FIGURE 13: A 10×10 synthetic grey scale image, object number 3 manually highlighted with dashed line to be contrasted from the black background.

the input image is withdrawn. Neurons are arranged in a one dimensional array which represents a linear mapping (line by line) of the 2D format of the grid. It clearly shows the synchronisation and desynchronisation of the temporal activities of neural oscillators in the network. While some neurons are active and their temporal activity is synchronised, other neurons are inactive and therefore their temporal activity is desynchronised (with those that are active and synchronised). At a given time, the active and synchronised neurons belong to the same object while those that are inactive belong to other different objects. The 3D graph provides a better illustration of the network activity.

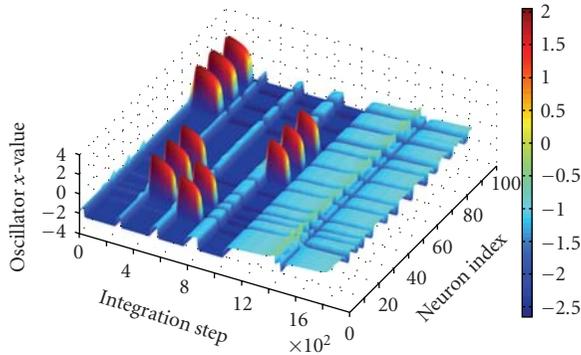


FIGURE 14: Temporal activity of the neural oscillators in the 10 by 10 grid. The z -axis represents the value of the x -unit of the neural oscillator (there are two units in an oscillator as described by the underlying differential equations, unit x and unit y). The x -axis represents the integration steps (only 1800 steps were shown here), and the y -axis represents the neurons' index (i.e., from neuron 1 which is associated with the top left corner pixel to neuron 100 which is associated with the bottom right corner pixel).

In order to further emphasise the segmentation of different objects based on the synchronisation and desynchronisation of different neurons in the 2-dimensional networks, the neurons which belong to the same object are grouped together and their temporal activity is displayed on the same axis as shown in Figure 15. The indexes of neurons which belong to the same object are first determined, and then the corresponding waveforms of those neurons are plotted on the same axis one after the other. Each neuron's waveform is separated from the previous neuron by an offset to avoid overlapping between two consecutive waveforms. Ordering the waveforms according to the objects they belong to highlights better which set of neurons are active at a given time, hence it indicates which object is detected at that time. For example at time T_1 (see Figure 15) the neurons that belong to object 4 are active while other neurons are inactive. At time T_2 , neurons that belong to object 1 are active and the remaining neurons (which belong to other objects and the background) are inactive. Similarly for time T_3 and time T_4 , the neurons belonging to object 2 and object 3 are active, respectively.

Using a 2D visualisation of the temporal activity of the neural oscillators of the network at selected points in time, Figure 16 illustrates further the segmented objects at the times shown in Figure 15. Time T_1 corresponds to object 4 (see Figure 13 for initial object labelling). The neural oscillators associated with the pixel of this object are active and also synchronised. The x -values of all neural oscillators in the grid are mapped to a grey scale image. The obtained 2D visualisation clearly shows at T_1 the emergence of object 4, while other neurons representing the remaining objects are inactive (small values) and therefore the corresponding objects fade away. Likewise at time T_2 , it is the neurons belonging to object 1 that are active and synchronised while other neurons are inactive. Similarly, at time T_3 and T_4 object 2 and object 3 have been temporally segmented, respectively.

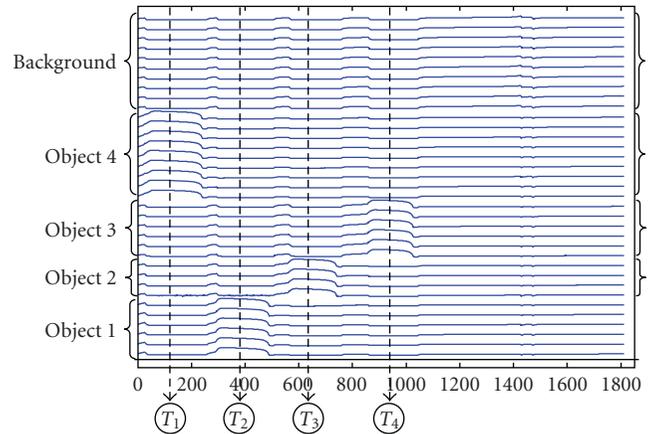


FIGURE 15: Temporal activity of neural oscillators ordered according to the objects they belong to. The x -axis represents integration steps and the y -axis represents neuron indexes. For example, the first six neurons belong to object 1, the next four neurons belong to object 2, the following six neurons belong to object 3, and then the following nine neurons represent object 4. Finally the remaining neurons (not all of them are drawn here for simplicity) represent the neurons mapped to the background pixels (there are $[(10 * 10) - (6 + 4 + 6 + 9)] = 75$ background neurons in total, only 10 neurons are shown).

5.2.2. Algorithmic Approach for Large Scale Images. For larger images, the activity of the neuron oscillators is calculated using an algorithmic approach as solving hundreds or thousands of systems of ordinary differential equations (ODE), each of which assigned to an image pixel, is prohibitively costly in terms of computing time and may not be feasible unless their discrete integration is implemented on dedicated hardware platform such as reconfigurable FPGAs (Field Programmable Gate Array) or GPUs (Graphics Processing Unit). The algorithmic approach is based on the neural oscillators dynamics on the cubic branches and at jumping points (LK, RK) between left and right branches (LB, RB). It produces the essential behaviours of coupled neural oscillators, namely the synchronisation and desynchronisation of their temporal activity as well as the exhibition of two time scales (i.e., the slow and fast motions on the left and right branches, resp.). The use of an algorithmic approach considerably reduces the computation time. Only the x value of an oscillator i is used and the algorithm is described in what follows.

Algorithm 1 is demonstrated on a 110-by-90-pixel image, which contains five chess pieces as shown in Figure 17. A network of 110 by 90 oscillators is built, and each neuron is stimulated with its corresponding pixel value. Each neural oscillator is connected to its four nearest neighbours, and the connecting weights are formed based on the image pixel values. Setting the coupling weights according to the expression $W_{ik} = I_m / (1 + |I_i - I_k|)$ results in oscillators with similar pixel values being assigned large weights while other oscillators belonging to different regions will be assigned weaker connections weights. The addition of the term 1 in the denominator is used to avoid a division by zero when both oscillators are assigned the same pixel values in which case the coupling term is assigned the maximum weight

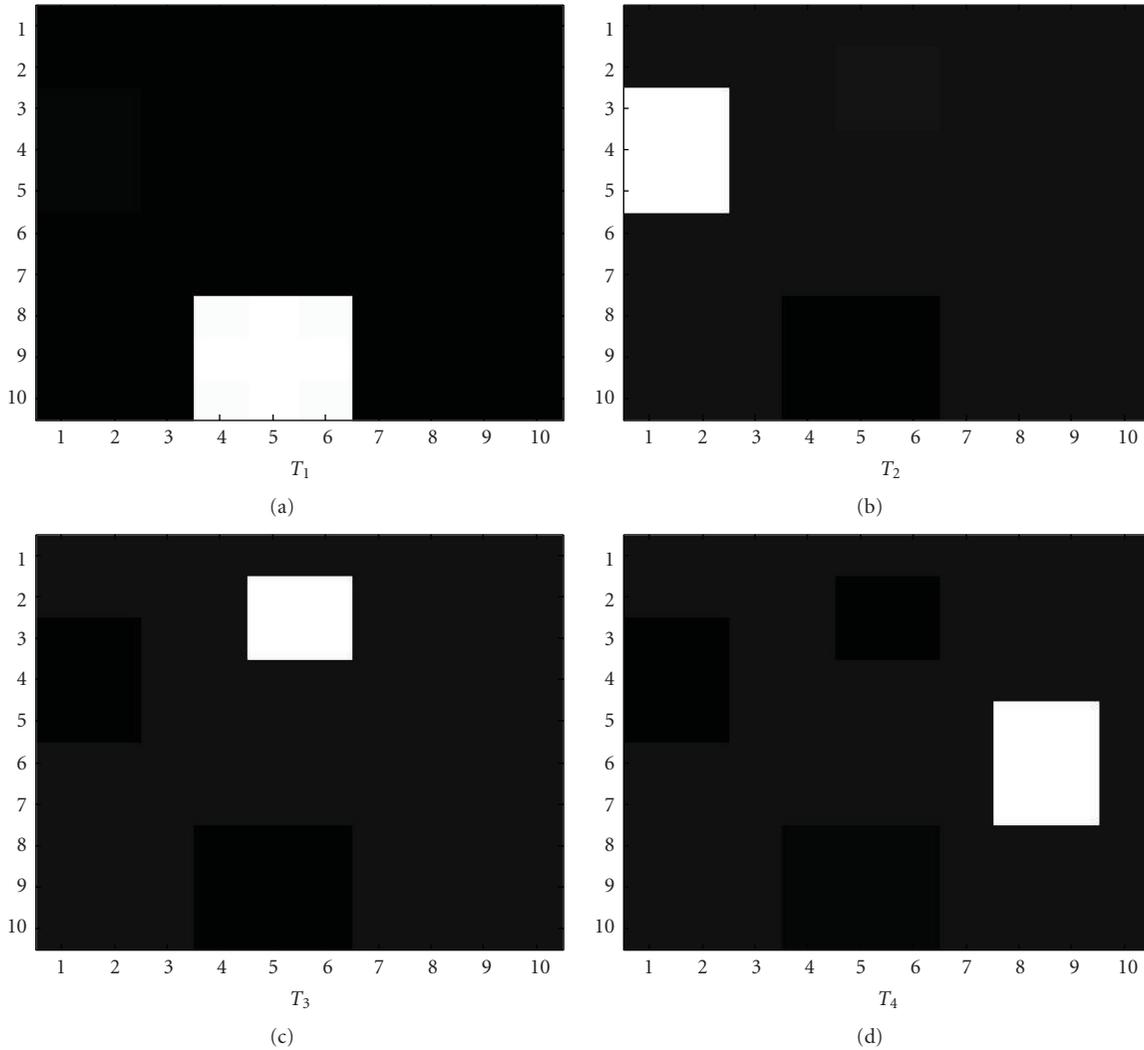


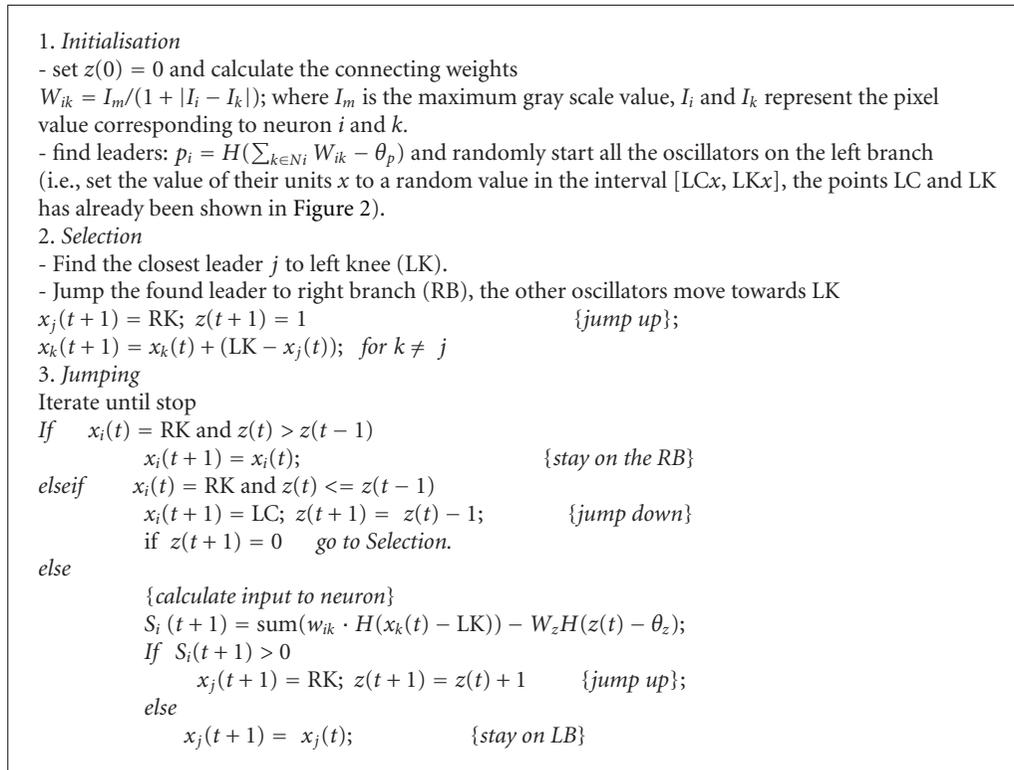
FIGURE 16: The value of the x -unit of each oscillator is shown in a 2D array where each element corresponds to one neural oscillator that is associated with a pixel of the image located at the same coordinates. The values of neural oscillators are selected at different points in time (T_1 , T_2 , T_3 , and T_4). These times correspond to those shown in Figure 15. The synchronisation of active neurons indicates which object is being segmented.

value. This will ensure that the leaders will propagate their activity to the oscillators in the same group representing the same object. As the network activity evolves, different objects (chess pieces) emerge one by one as shown in Figure 17. The emergence of each object is achieved through the synchronisation of the activities of neurons belonging to the same object (images (a) to (d)). Image (f) represents the image background.

6. Application of Neural Oscillators to Colour Image Segmentation

A similar network of oscillators is applied to colour images where the HSV (hue, saturation, value) colour space is used and a self-organising map-based colour reduction method is employed. The colour reduction method allows the extension of the applicability of networks of oscillators

to colour image segmentation. As each neuron in the grid of oscillators receives a one-dimensional input value, there is a need for a mapping method between RGB triplets of an input colour image and the corresponding feature values where similarity measurements are taken into account colour. To the best of our knowledge, this is the first time such neural models are applied to colour image segmentation. The HSV colour space, a linear transform of the RGB (red, green, blue) space [3], offers the advantage of specifying colours in an extremely intuitive manner as the luminance component is separated from the chrominance (i.e., colour) information [2, 58]. The latter feature makes this colour space more suitable for digital image processing applications as it is easy to select a desired colour (hue component) and to then modify it slightly by adjustment of its saturation and intensity [58]. In addition, the components of the HSV colour space are normalised, which makes it suitable for the



ALGORITHM 1

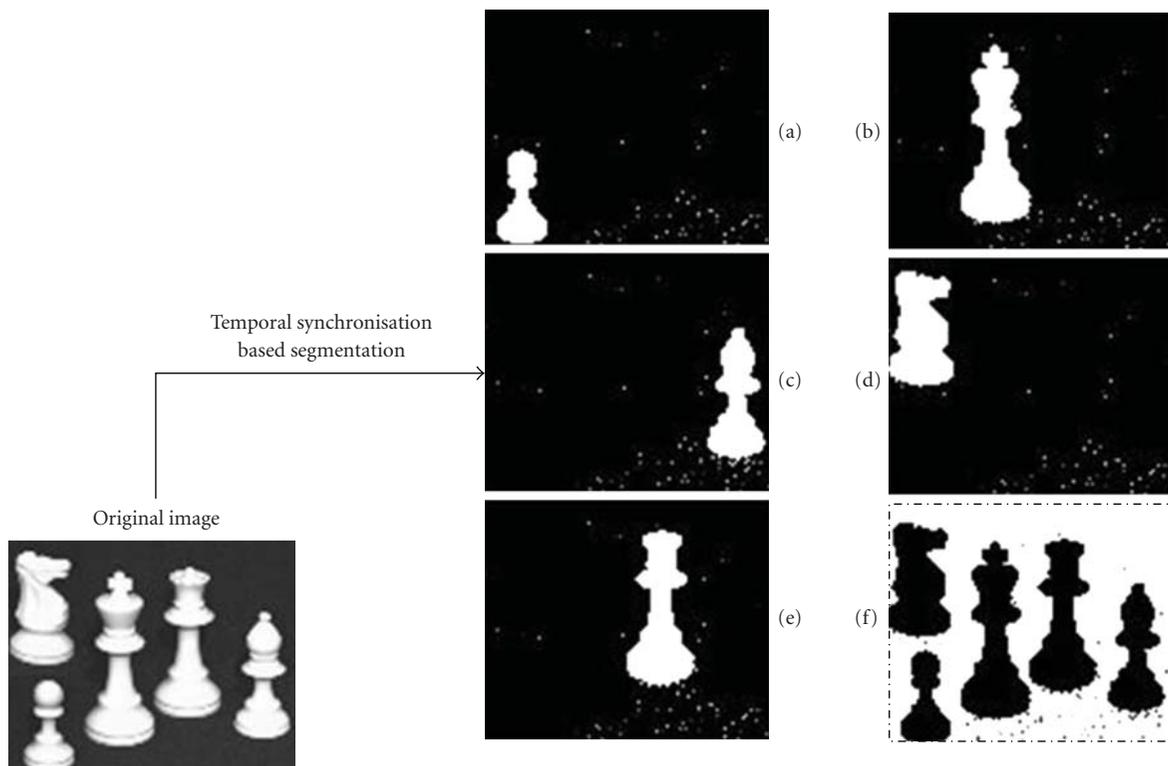


FIGURE 17: Segmented objects from 110×90 image containing five chess pieces. Left: original image. (a) to (e) Segmented objects. (f) The image background.

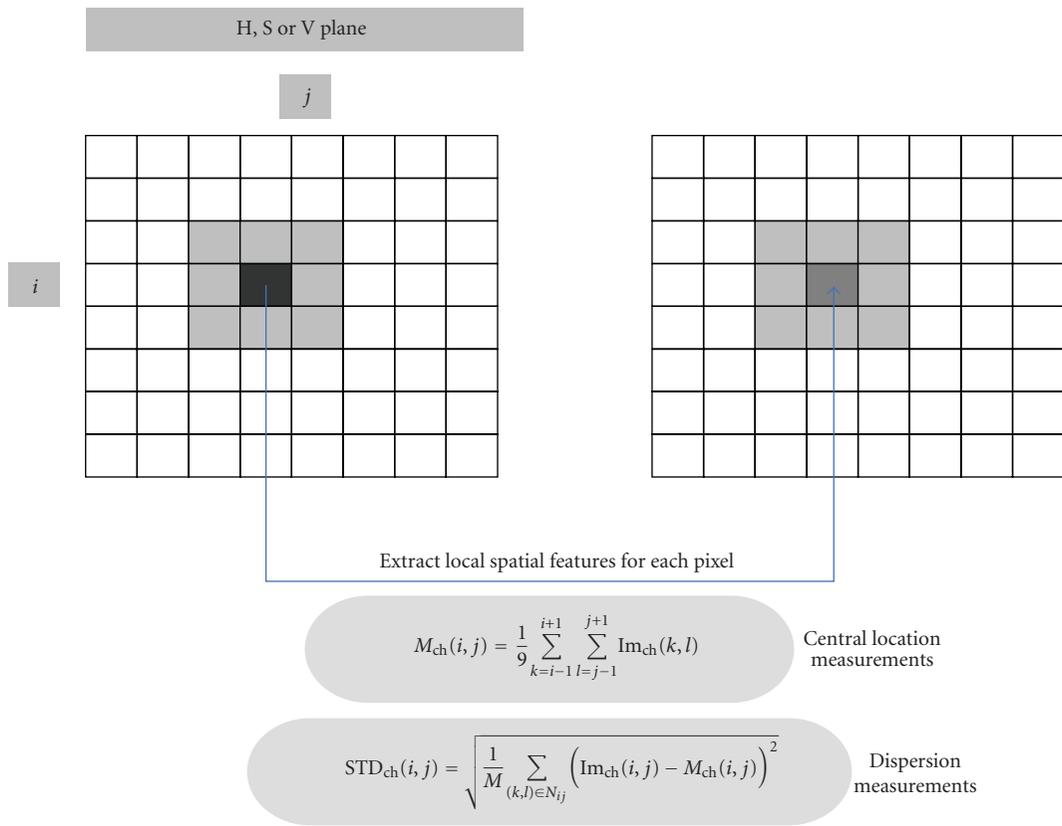


FIGURE 18: Extraction of local spatial features. The mean and standards deviation are calculated for each pixel in the three chromatic planes. Hence, each pixel in the image results in six different spatial features. These spatial features are calculated over a 3 by 3 window centred at the target pixel.

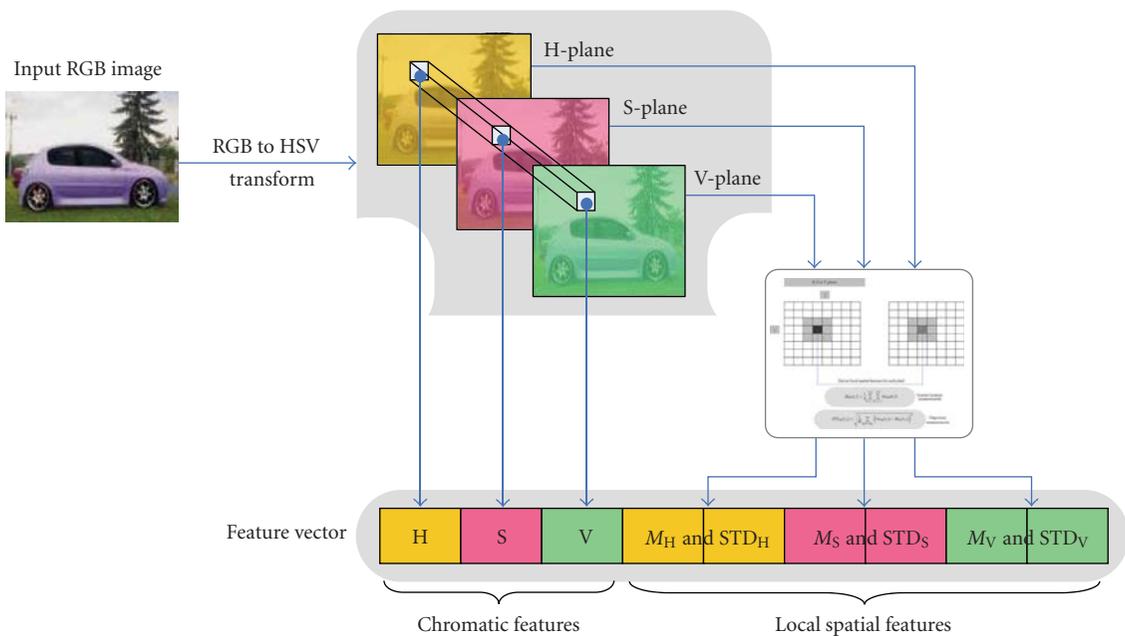


FIGURE 19: Feature vector for each pixel, using both spatial and chromatic characteristics.

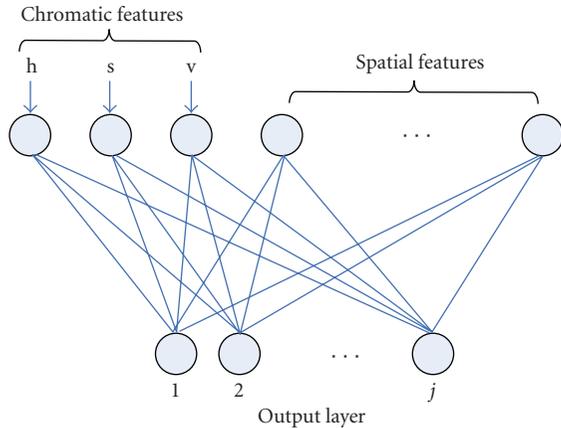


FIGURE 20: Kohonen's SOM used for colour reduction and image transformation. The chromatic components and spatial features are used as inputs.

colour reduction method and the system of colour image segmentation developed in this work.

The colour reduction method exploits both the chromatic features of pixels as well as their spatial characteristics such as how the colour of each pixel is related to the colours of its neighbouring pixels. It is the newly obtained reduced space of colours that is eventually used by a network of locally connected neural oscillators to achieve the segmentation of existing objects.

6.1. Colour Reduction Using Kohonen Self-Organising Maps. Given an RGB input image with a certain number of colours (16 M colours with a 24-bit resolution), the final reduced set of colours is automatically selected using a Kohonen self-organising map (SOM) which is an unsupervised neural network devised by Kohonen [59]. It is based on competitive learning and is a topology preserving map [59, 60], a feature that is highly desired for high dimensional data visualisation and reduction.

In this work, the chromatic features (i.e., the HSV components) of each pixel are combined with additional spatial features extracted from neighbouring pixels. The mean and standard deviations are used here; however there is no restriction on the type of spatial features that could be used. The goal is to map the HSV colour space into a one-dimensional space of colour indexes that can be used by the network of oscillators. Once the set of feature vectors of a given image is constructed, a SOM is built and trained according to Kohonen's competitive learning rule [59]. The main characteristic of this mapping is that it takes into account the chromatic and spatial similarities between neighbouring pixels. Eventually, the input image, with a larger set of colours, is transformed into another image with a limited number of colours while the spatial characteristics are being maintained.

6.2. RGB to HSV Colour Space Transform. The RGB colour space $[0, 255]^3$ is mapped into the HSV colour space $[0, 1]^3$

where the input image is then represented. The RGB to HSV transform is performed using the Matlab function *rgb2hsv* routine. A new representation is obtained where each image pixel is represented by a triplet (H, S, V) which determines its chromatic features. The system assumes that the input image is in HSV format; however this is not a restriction as it is always possible to map an image from one colour space to another.

6.3. Chromatic and Spatial Feature Extraction. The final feature vector which describes each image pixel consists of its chromatic features combined with its local spatial features calculated for each chromatic plane. The local spatial characteristics chosen in this work are represented by the mean and the standard deviation of the pixel values within a window centred at the pixel being considered. They are calculated as follows.

6.3.1. Central Location Measurements (Mean Values). This spatial feature is based on statistical measurements of the central location of each pixel within the sample points selected from a given window, namely, the mean value which is given by the following formula:

$$M_{ch}(i, j) = \frac{1}{M} \sum_{(k,l) \in N_{ij}} Im_{ch}(i, j), \quad (9)$$

where ch represent the chromatic plane being considered $ch \in \{h, s, v\}$, i, j are the pixel coordinates for which the spatial feature is being calculated, N_{ij} is the neighbourhood of the pixel (i, j) (i.e., a window centred at the pixel (i, j)), and M is the total number of pixels within that window in the input image Im (see illustration in Figure 18).

6.3.2. Dispersion Measurements (Standard Deviation Values). Likewise, the standard deviation is computed for each chromatic component using the following formula:

$$STD_{ch}(i, j) = \sqrt{\frac{1}{M} \sum_{(k,l) \in N_{ij}} (Im_{ch}(i, j) - M_{ch}(i, j))^2}. \quad (10)$$

In this work, a window of size 3×3 is used and the image is padded with zeros before the spatial characteristics are computed.

As a result, each pixel is described by nine values representing the chromatic and spatial features as illustrated in Figure 19. It is the size of the feature vector that will dictate the number of inputs of the Kohonen self-organising network. A set of these feature vectors will be created and used to build and train a Kohonen self-organising map as explained in Section 6.4.

6.4. Building and Training of Kohonen Self-Organising Network. A Kohonen SOM is designed, where the inputs represent the chromatic and the local spatial features of a pixel as calculated in the previous section. While the number of inputs is dictated by the dimensionality of the feature space, the size of the output layer represents the

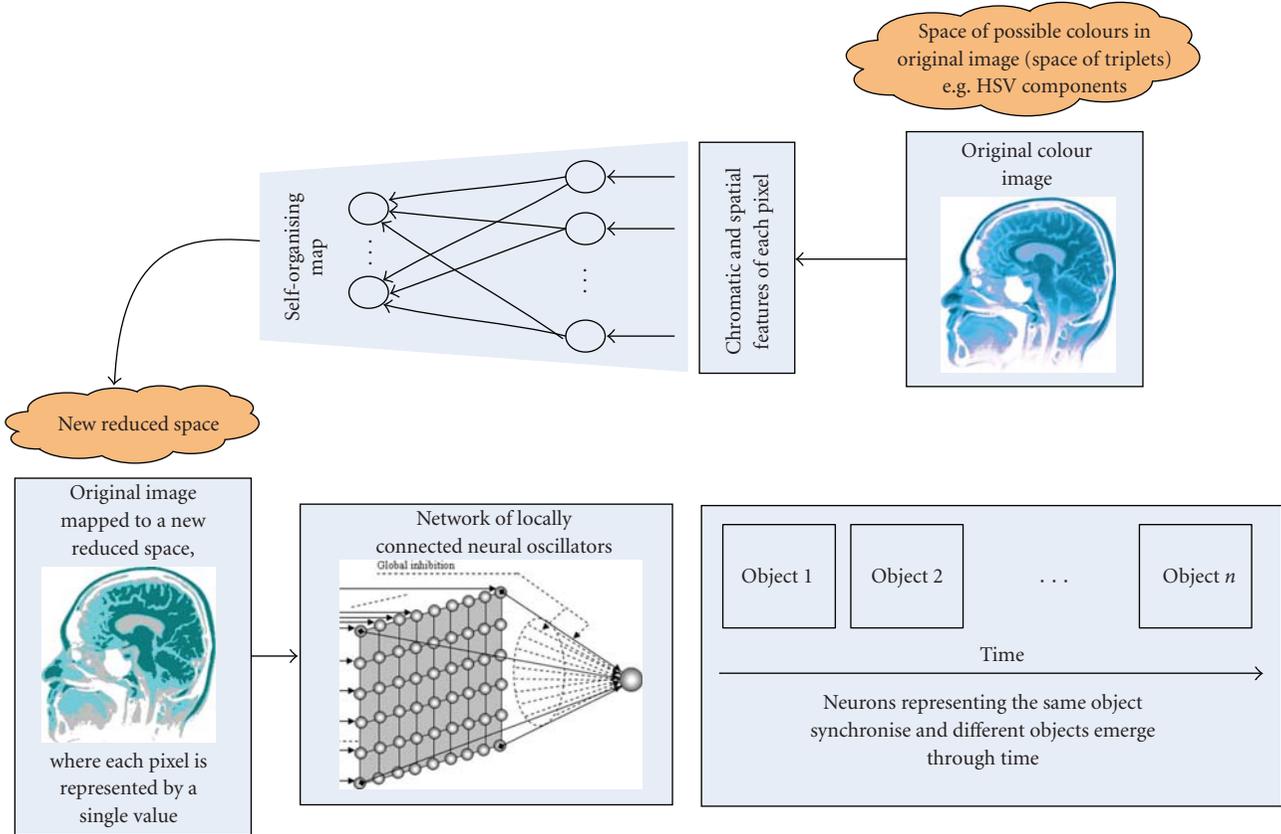


FIGURE 21: Neuro-inspired framework for colour image segmentation. The output from a self-organising maps-based colour reduction stage is fed to a network of locally excited globally inhibited neural oscillators and existing objects are segmented through temporal synchronisation of neurons belonging to the same coherent areas.

number of representative colours in the reduced colour space (see Figure 20). A subset of feature vectors is randomly selected to form the training set, the SOM network is then trained, and its final weights are adjusted using the Kohonen competitive learning rule. It is important to note that for each new image, a SOM is created anew and its weights are trained, and adapted to the new image. Hence, the mapping between the original colour space and the reduced one is updated and adapted to the content of the image to be segmented.

6.5. *Mapping.* After training, the new reduced set of colours is represented by the final weights of the output neurons which are then used to map an input image into the newly created colour space. The new image contains only the representative colours as learned by the Kohonen self-organising map. Each pixel in the image is assigned its most similar colour defined by the trained SOM.

6.6. *A Neurosystem for Colour Image Segmentation.* The combination of the dynamics of networks of neural oscillators and the self-organising map-based colour reduction approach, described in the previous section, results in a neurosystem for colour image segmentation. The original

image is first converted into the HSV space, then the representative colours in a given image are automatically selected using the unsupervised Kohonen self-organising map, which is trained anew for each input image. The resulting image with a reduced number of colours is then segmented using the temporal correlation of neural oscillators. An illustration of the proposed framework for colour image segmentation based on the combination of networks of neural oscillators and the SOM-based colour reduction method is illustrated in Figure 21.

6.7. *MATLAB Simulation Results.* The proposed approach has been implemented in Matlab and demonstrated on 24-bit bitmap colour images (16 million colours as original space) representing real world scenes. The first image represents a car and contains 130×80 pixels (see Figure 22 bottom right corner), the second image represents a hand and has 100×129 pixels (see Figure 23), and the third image represents a cup and has 128×128 pixels (see Figure 24). The input images are originally represented in the RGB space. A transformation to the HSV space is carried out using the Matlab *rgb2hsv* function which comes with the image processing toolbox. A Kohonen's self-organising network is then created and trained and used for colour reduction,

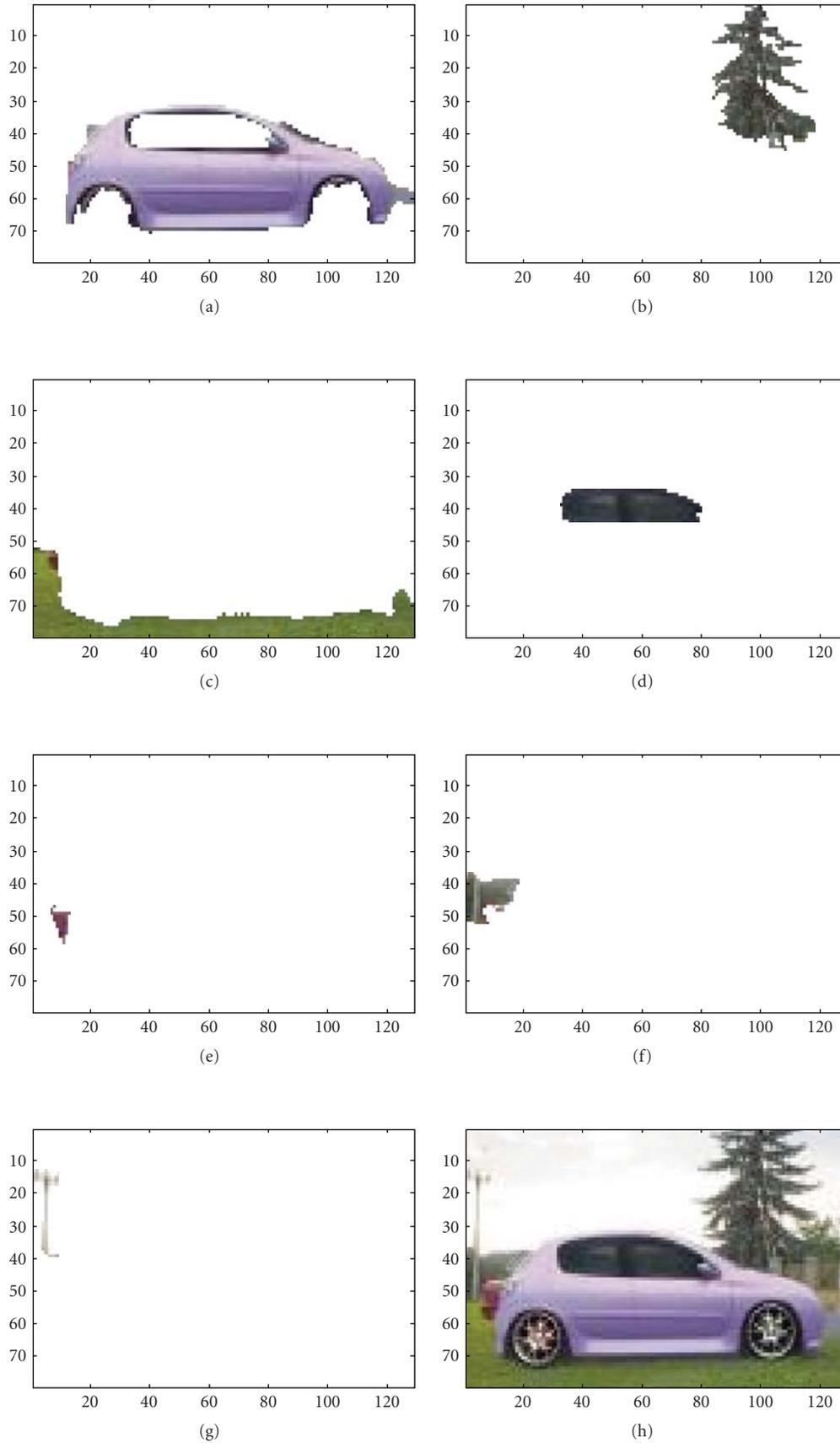


FIGURE 22: Segmentation of a sample colour image (130 by 80 pixels) into different regions.

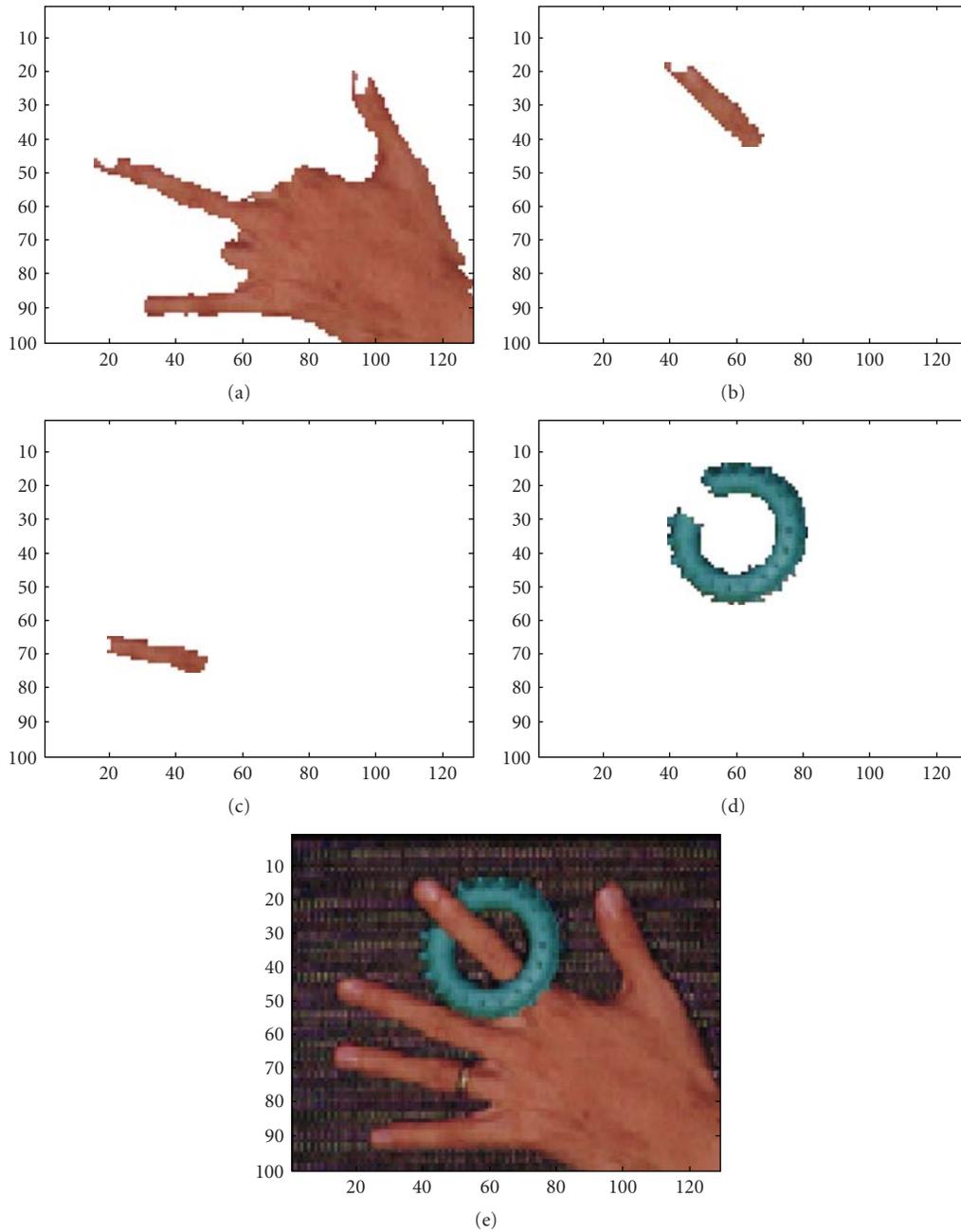


FIGURE 23: Segmentation of another sample colour image (100 by 129 pixels) into different regions. Fingers are separated due to the surrounding rings (the big blue ring and the small golden one). This is in accordance with the definition of an object as mentioned in early sections, that is, a coherent region with similar pixels.

where its final weights are used to transform an input image into a new image with pixel values mapped to the newly obtained reduced space. Finally, a network of neural oscillators (with a size equal to the input image size) is created, where the pixel values of the transformed image are used as external inputs to neural oscillators. The coupling weights are also determined according to the newly obtained pixel values. The segmentation results are illustrated in Figures 22–24 where the main segmented objects are shown separately along with the original image.

7. Discussion

A subjective evaluation is carried out to assess the segmentation quality of the developed system. It is based on the human visual assessment, the most common practice for evaluating the effectiveness of a segmentation system. Figures 22–24 show that the segmented images are of an acceptable quality for use in computer-aided diagnosis, image understanding, or object recognition schemes. Although the evaluation remains subjective, this satisfactory assessment is based on

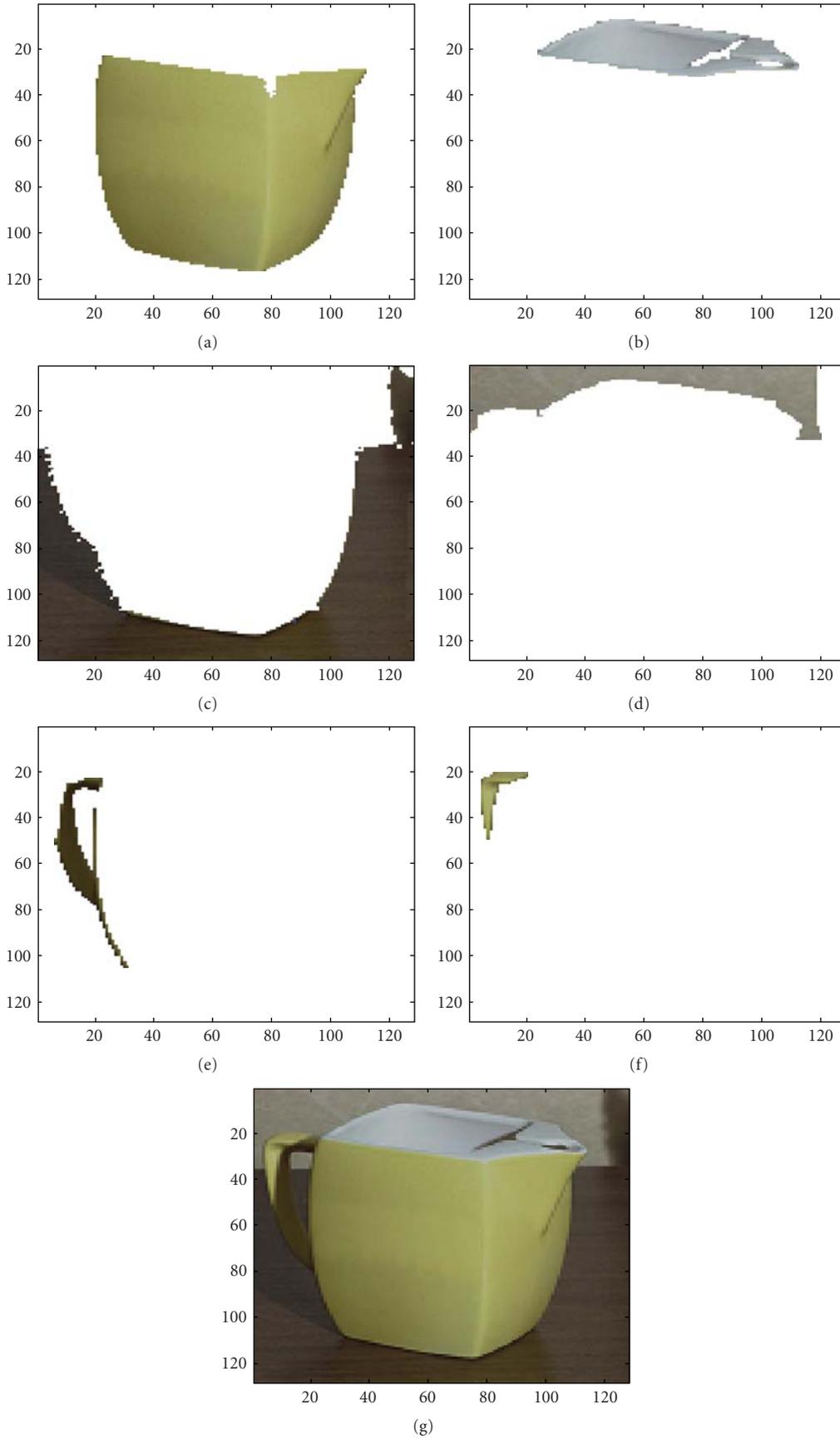


FIGURE 24: Segmentation of a third sample colour image (128 by 128 pixels) into different regions.

the clear separation of the existing objects within the input image. It can be seen from Figure 17 how all existing objects (chess pieces and the background) are accurately segmented. Likewise, the segmentation of the synthetic image using the step-by-step integration of neural oscillators resulted in a perfect separation of the existing rectangles. The problem is, however, trivial in this case; yet it helps demonstrate and validate the underlying principle of object segmentation using neural oscillators and objectively assess its quality. The assessment is objective in this example since a synthetic image is used as the ground truth segmentation. The segmentation of the colour images in Figures 22–24 clearly shows that the segmentation of the coherent areas (i.e., objects) in the image (car frame, windows/glass, tree, grass, hand, ring, cup, etc.) is fairly satisfactory. The evaluation of segmentation quality remains an open problem in the field of image segmentation. The absence of accurate and automatic objective evaluation methods makes it difficult to assess whether one algorithm produces better segmentation quality than another. The objective of this paper is to demonstrate the applicability of bioinspired neural oscillators to colour image segmentation and the visual quality of the obtained results clearly demonstrate that the principle is effective. A comparison with all existing algorithmic and machine learning methods is out of the scope of this paper. However, the approach presented in this paper can still be compared to other approaches in terms of its underlying computing principles, advantages and disadvantages. First of all, the computing paradigm of this system can be likened to a region growing approach as more pixels are grouped within a given region (grown from a chosen seed) when their features satisfy some specified criteria. However, the system in this work remains different due to the inherent dynamics of synchronisation and desynchronisation of neurons temporal activity and also the colour reduction method which allows automatic selection of a reduced number of the most relevant representative colours of the input image. The other distinguishing characteristic of this approach is that the reduced set of representative colours are created anew for each input image; that is, it is adaptable to each new input image, as it takes into account both the chromatic and local spatial features which are specific to each image being handled. The other apparent differences and advantages of this system, are that they offer a biologically inspired approach (whose modules are not necessarily all biologically plausible) which builds on the strengths of neural architectures and the computing principles of the brain, which is undoubtedly the *only* segmentation system that is perfectly working. Another advantage resides in the parallelism and distributed computing of this approach as all the states of oscillators are computed locally and in parallel which makes it more suitable for dedicated hardware implementation (such as VLSI, FPGAs, and/or GPUs) in order to reduce the computation time and achieve real-time computations. This is a very desirable feature for segmenting large scale image databases. Moreover, this method is not specific to particular types of objects and requires less human intervention as opposed to existing segmentation techniques where the user needs to specify initially the number of objects

to be segmented. Yet, this approach can be extended to more specific domains where available a priori knowledge can be exploited to calculate the most relevant features.

8. Conclusion and Future Work

A biologically inspired neural model, called neural oscillators, has been investigated and its potential application to solve real world problems, namely, colour image segmentation, has been explored. Also a new system for colour image segmentation was presented. The biological relevance of neural oscillators has been highlighted and their dynamics examined through simulations and analysis of single and coupled neurons as well as networks of neurons. Analysis of the behaviours of this type of neurons was carried out in the time domain as well as in the phase plane, and an investigation of their application to segmentation of grey scale and colour images was presented. The problem of feature binding was discussed and its connection with neural oscillators was highlighted. Also, a discussion of the established link between the binding problem and computer vision was provided along with the role of temporal correlation in feature binding. The main contribution of this work consists of the exploitation of the possibilities of computing and learning with neural oscillators in solving real world engineering problems. To the best of the author's knowledge, this is the first attempt to apply networks of neural oscillators to the segmentation of colour images. The segmentation quality was visually assessed and a comparison between the proposed approach and classical image segmentation counterparts was carried out. Segmentation results demonstrate the emergence of neural oscillators, combined with self-organising, as an efficient alternative approach for colour image segmentation; it offers the desirable feature of parallel computing which can be further exploited for hardware implementations. Mapping the system onto dedicated hardware platforms will be a desirable extension of this work as it will allow real-time segmentation of large scale colour images and datasets.

References

- [1] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.
- [2] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, McGraw-Hill, New York, NY, USA, 1995.
- [3] R. C. Gonzalez and E. W. Richard, *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.
- [4] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, & Image Processing*, vol. 29, no. 1, pp. 100–132, 1985.
- [5] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.
- [6] W. Skarbek and A. Koschan, "Colour image segmentation: a survey," Tech. Rep., Institute for Technical Informatics, Technical University of Berlin, Berlin, Germany, October 1994.

- [7] V. Zharkova, S. Ipson, J. Abouardham, and B. Bentley, "Survey of image processing techniques, EGSO internal deliverable," Tech. Rep. EGSO-5-D1_F03-20021029, 2002.
- [8] C. von der Malsburg and W. Schneider, "A neural cocktail-party processor," *Biological Cybernetics*, vol. 54, no. 1, pp. 29–40, 1986.
- [9] C. von der Malsburg and J. Buhmann, "Sensory segmentation with coupled neural oscillators," *Biological Cybernetics*, vol. 67, no. 3, pp. 233–242, 1992.
- [10] W. Singer and C. M. Gray, "Visual feature integration and the temporal correlation hypothesis," *Annual Review of Neuroscience*, vol. 18, pp. 555–586, 1995.
- [11] M. Ursino and G.-E. La Cara, "Modeling segmentation of a visual scene via neural oscillators: fragmentation, discovery of details and attention," *Network: Computation in Neural Systems*, vol. 15, no. 2, pp. 69–89, 2004.
- [12] J. Grasman, *Asymptotic Methods for Relaxation Oscillations and Applications*, Springer, New York, NY, USA, 1987.
- [13] F. Verhulst, *Nonlinear Differential Equations and Dynamical Systems*, Springer, Berlin, Germany, 1996.
- [14] B. van der Pol, "On relaxation oscillations," *Philosophical Magazine*, vol. 2, no. 11, pp. 978–992, 1926.
- [15] A. V. Hill, "Wave transmission as the basis of nerve activity," *Cold Spring Harbor Symposia on Quantitative Biology*, vol. 1, pp. 146–151, 1933.
- [16] B. van der Pol and J. van der Mark, "The heartbeat considered as a relaxation oscillation, and an electrical model of the heart," *Philosophical Magazine*, vol. 6, pp. 763–775, 1928.
- [17] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, no. 4, pp. 500–544, 1952.
- [18] R. FitzHugh, "Impulses and physiological states in models of nerve membrane," *Biophysical Journal*, vol. 1, pp. 445–466, 1961.
- [19] J. S. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon," *Proceedings of the IRE*, vol. 50, pp. 2061–2070, 1962.
- [20] E. Mayeri, "A relaxation oscillator description of the burst generating mechanism in the cardiac ganglion of the lobster, *Homarus americanus*," *Journal of General Physiology*, vol. 62, no. 4, pp. 473–488, 1973.
- [21] H. R. Wilson and J. D. Cowan, "Excitatory and inhibitory interactions in localized populations of model neurons," *Biophysical Journal*, vol. 12, no. 1, pp. 1–24, 1972.
- [22] H. R. Wilson and J. D. Cowan, "A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue," *Kybernetik*, vol. 13, no. 2, pp. 55–80, 1973.
- [23] D. J. Pinto, J. C. Brumberg, D. J. Simons, and G. B. Ermentrout, "A quantitative population model of whisker barrels: re-examining the Wilson-Cowan equations," *Journal of Computational Neuroscience*, vol. 3, no. 3, pp. 247–264, 1996.
- [24] H. R. Wilson, "Simplified dynamics of human and mammalian neocortical neurons," *Journal of Theoretical Biology*, vol. 200, no. 4, pp. 375–388, 1999.
- [25] S. Campbell and D. Wang, "Synchronization and desynchronization in a network of locally coupled Wilson-Cowan oscillators," *IEEE Transactions on Neural Networks*, vol. 7, no. 3, pp. 541–554, 1996.
- [26] C. Morris and H. Lecar, "Voltage oscillations in the barnacle giant muscle fiber," *Biophysical Journal*, vol. 35, no. 1, pp. 193–213, 1981.
- [27] C. M. Gray, P. König, A. K. Engel, and W. Singer, "Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties," *Nature*, vol. 338, no. 6213, pp. 334–337, 1989.
- [28] R. Eckhorn, R. Bauer, W. Jordan, et al., "Coherent oscillations: a mechanism of feature linking in the visual cortex?" *Biological Cybernetics*, vol. 60, no. 2, pp. 121–130, 1988.
- [29] W. Singer and C. M. Gray, "Visual feature integration and the temporal correlation hypothesis," *Annual Review of Neuroscience*, vol. 18, pp. 555–586, 1995.
- [30] D. G. Aronson, G. B. Ermentrout, and N. Kopell, "Amplitude response of coupled oscillators," *Physica D*, vol. 41, no. 3, pp. 403–449, 1990.
- [31] D. Cairns, R. Baddeley, and L. Smith, "Constraints on synchronizing oscillator networks," *Neural Computation*, vol. 5, no. 2, pp. 260–266, 1993.
- [32] J. Grasman and M. J. W. Jansen, "Mutually synchronized relaxation oscillators as prototypes of oscillating systems in biology," *Journal of Mathematical Biology*, vol. 7, no. 2, pp. 171–197, 1979.
- [33] R. E. Plant, "A fitzhugh differential-difference equations modeling recurrent neural feedback," *SIAM Journal on Applied Mathematics*, vol. 40, no. 1, pp. 150–162, 1981.
- [34] S. Grossberg and D. Somers, "Synchronized oscillations during cooperative feature linking in a cortical model of visual perception," *Neural Networks*, vol. 4, no. 4, pp. 453–466, 1991.
- [35] D. Somers and N. Kopell, "Waves and synchrony in networks of oscillators of relaxation and non-relaxation type," *Physica D*, vol. 89, no. 1-2, pp. 169–183, 1995.
- [36] D. Terman and D. Wang, "Global competition and local cooperation in a network of neural oscillators," *Physica D*, vol. 81, no. 1-2, pp. 148–176, 1995.
- [37] C. von der Malsburg, "The correlation theory of brain function," Internal Rep. 81-2, Max-Planck-Institute for Biophysical Chemistry, 1981.
- [38] C. von der Malsburg and W. Schneider, "A neural cocktail-party processor," *Biological Cybernetics*, vol. 54, no. 1, pp. 29–40, 1986.
- [39] W. A. Phillips and W. Singer, "In search of common foundations for cortical computation," *Behavioral and Brain Sciences*, vol. 20, no. 4, pp. 657–722, 1997.
- [40] A. Keil, M. M. Müller, W. J. Ray, T. Gruber, and T. Elbert, "Human gamma band activity and perception of a Gestalt," *Journal of Neuroscience*, vol. 19, no. 16, pp. 7152–7161, 1999.
- [41] J. C. Prechtl, "Visual motion induces synchronous oscillations in turtle visual cortex," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 91, no. 26, pp. 12467–12471, 1994.
- [42] K. MacLeod and G. Laurent, "Distinct mechanisms for synchronization and temporal patterning of odor-encoding neural assemblies," *Science*, vol. 274, no. 5289, pp. 976–979, 1996.
- [43] V. N. Murthy and E. E. Fetz, "Coherent 25- to 35-Hz oscillations in the sensorimotor cortex of awake behaving monkeys," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, no. 12, pp. 5670–5674, 1992.
- [44] A. K. Engel, P. König, A. K. Kreiter, and W. Singer, "Interhemispheric synchronization of oscillatory neuronal responses in

- cat visual cortex,” *Science*, vol. 252, no. 5010, pp. 1177–1179, 1991.
- [45] D. Somers and N. Kopell, “Rapid synchronization through fast threshold modulation,” *Biological Cybernetics*, vol. 68, no. 5, pp. 393–407, 1993.
- [46] J. Ontrup, H. Wersing, and H. Ritter, “A computational feature binding model of human texture perception,” *Cognitive Processing*, vol. 5, pp. 32–44, 2004.
- [47] A. L. Roskies, “The binding problem,” *Neuron*, vol. 24, no. 1, pp. 7–9, 1999.
- [48] C. M. Gray, “The temporal correlation hypothesis of visual feature integration: still alive and well,” *Neuron*, vol. 24, no. 1, pp. 31–47, 1999.
- [49] W. Singer, “Neuronal synchrony: a versatile code for the definition of relations?” *Neuron*, vol. 24, no. 1, pp. 49–65, 1999.
- [50] G. Calvert, C. Spence, and B. E. Stein, *The Handbook of Multisensory Processes*, MIT Press, Cambridge, Mass, USA, 2004.
- [51] A. Revonsuo and J. Newman, “Binding and consciousness,” *Consciousness and Cognition*, vol. 8, no. 2, pp. 123–127, 1999.
- [52] J. M. Wolfe and K. R. Cave, “The psychophysical evidence for a binding problem in human vision,” *Neuron*, vol. 24, no. 1, pp. 11–17, 1999.
- [53] H. B. Barlow, “Single units and sensation: a neuron doctrine for perceptual psychology?” *Perception*, vol. 1, no. 4, pp. 371–394, 1972.
- [54] R. Desimone, T. D. Albright, C. G. Gross, and C. Bruce, “Stimulus-selective properties of inferior temporal neurons in the macaque,” *Journal of Neuroscience*, vol. 4, no. 8, pp. 2051–2062, 1984.
- [55] D. I. Perrett, E. T. Rolls, and W. Caan, “Visual neurones responsive to faces in the monkey temporal cortex,” *Experimental Brain Research*, vol. 47, no. 3, pp. 329–342, 1982.
- [56] S. Zeki, *A Vision of the Brain*, Blackwell Scientific, Oxford, UK, 1993.
- [57] W. M. Usrey and R. C. Reid, “Synchronous activity in the visual system,” *Annual Review of Physiology*, vol. 61, pp. 435–456, 1999.
- [58] A. Ford and A. Roberts, *Colour Space Conversions*, Westminster University, London, UK, 1998.
- [59] T. Kohonen, *Self-Organizing Maps*, Springer, Berlin, Germany, 1995.
- [60] S. Haykin, *Neural Networks and Learning Machines*, Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2008.

Research Article

Evaluation of Data Quality and Drought Monitoring Capability of FY-3A MERSI Data

Daxiang Xiang,¹ Liangming Liu,¹ Qiao Wang,² Na Yang,¹ and Tao Han³

¹ School of Remote Sensing and Information Engineering, Wuhan University, Wuhan Hubei 430079, China

² Network & Information Center, Changjiang Water Resources Commission, Wuhan Hubei 430010, China

³ Key Laboratory of Arid Climatic Change and Reducing Disaster of Gansu Province and Key Open Laboratory of Arid Climatic Change and Reducing Disaster of CMA, Institute of Arid Meteorology, Lanzhou Gansu 730020, China

Correspondence should be addressed to Daxiang Xiang, daxiangx@163.com

Received 22 September 2009; Accepted 18 November 2009

Academic Editor: Djamel Bouchaffra

Copyright © 2010 Daxiang Xiang 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.

FY-3A is the second Chinese Polar Orbital Meteorological Satellite with global, three-dimensional, quantitative, and multispectral capabilities. Its missions include monitoring global disasters and environment changes. This study describes some basic parameters and major technical indicators of the FY-3A and evaluates data quality and drought monitoring capability of the *Medium-Resolution Imager* (MERSI) onboard the FY-3A. Data obtained with the MERSI was compared with that of the *MODerate-resolution Imaging Spectroradiometer* (MODIS), imaged at the same time period and geographic zone. In addition, the *Temperature/Vegetation Drought Index* (TVDI), a highly accurate and stable monitoring model, was used to monitor drought condition with MERSI and MODIS sensors. It is found in the study that the relative accuracy of data, obtained with these two devices, was consistent with the acceptable overall accuracy of 93.8. Furthermore, spatial resolution of MERSI is superior as compared to that of MODIS. Therefore, FY-3A MERSI can serve a reliable and new data source for drought monitoring.

1. Introduction

1.1. Drought Monitoring. Drought is a normal, recurrent feature of climate. It occurs almost everywhere, although its features vary from region to region. In the most general sense, drought originates from a deficiency of precipitation over an extended period of time, resulting in a water shortage for some activity, group, or environmental sector (NDMC define). Consequently, it is required to demonstrate the distribution and degree of drought condition timely, which is crucial for drought warning and resisting effectively [1–3].

At present, remote sensing methods for drought monitoring are mainly classified into four categories: Vegetation Index-based [4, 5], Temperature-based [6], Vegetation and Temperature-based [7], and Cloud-based [8]. The representative indices include *Vegetation Supply Water Index* (VSWI) [9], *Temperature/Vegetation Dryness Index* (TVDI) [10, 11], *Apparent Thermal Inertia Index* (ATI) [12, 13], and *Cloud Parameters Index* (CPI) [14]. As FY-3A satellite is a polar-orbital, its temporal resolution is hard to meet the requirements of cloud-based method, so CPI is not utilized

in this study. ATI requires the index of temperature variation between day and night. However, polar-orbital satellites cannot provide real-time temperature; consequently ATI method is not suitable for FY-3A to monitor drought. VSWI and TVDI need only two parameters—*Normal Difference Vegetation Index* (NDVI) and *Land Surface Temperature* (LST), which can be calculated from MERSI data. Meanwhile, taking the superiority of TVDI into account, it is used in this study.

1.2. FY-3A MERSI. The FY-3A satellite is the second Chinese polar orbital meteorological satellite which provides global, three-dimensional, high-quantified, and multispectral remote sensed data. The satellite weighs 2400 kg, and has in-orbit dimensions of $4.46 \times 10 \times 3.79$ meters. Its orbit altitude is 836.4 km, with an obliquity of 98.753 degrees and an orbit time of 101.496 minutes. Its designed life-span is over 2 years. FY-3A has refined the sensor technology, which is multisensor, microwave, hundred-meter resolution, and global data reception rather than single-sensor, optical

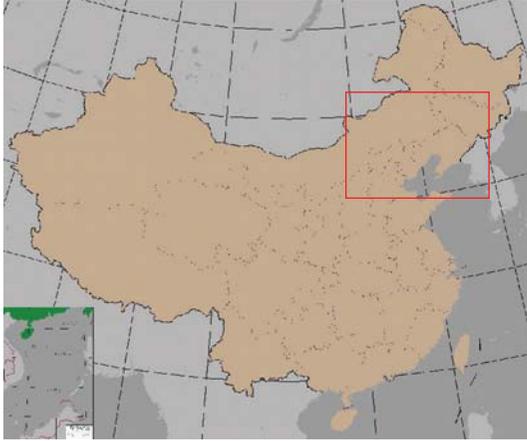


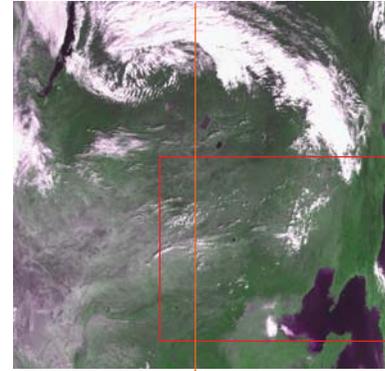
FIGURE 1: Study area in North China.

only, kilo-meter resolution, and domestic data reception in the field of meteorological satellite. FY-3A missions include monitoring natural disasters and environmental changes all over the world and offering meteorological information for agriculture. Since, drought monitoring has naturally become one of the important applications of FY-3A. Therefore, drought monitoring capability of FY-3A deserves a comprehensive evaluation. Using MODIS results obtained at the same time period as reference, data quality, and drought monitoring capability of MERSI is analyzed in the study (CMA).

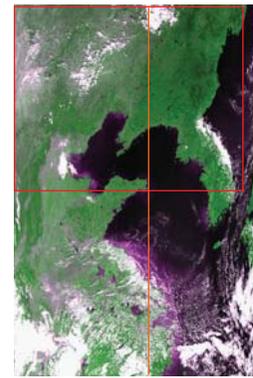
With multispectral and high-resolution features, MERSI can detect atmosphere, land, and ocean through the reflection of visible channels and thermal infrared radiation. There are three channels located in the water vapor absorption window (0.905~0.980 μm) which increase detection capability for atmospheric water vapor, in particular water vapor at low levels. Visible bands with spatial resolution of 250 meters can be utilized to derive NDVI and thermal bands for LST. Hence, MERSI has an enormous potential for drought monitoring. Details about some spectral properties of MERSI are listed in Table 1.

It can be found in Table 1 that FY-3A MERSI has red, near-infrared, and thermal infrared channels with spatial resolution of 250 meters, so it can provide drought monitoring information of 250 meters. This should be a great improvement of meteorological satellite data in spatial resolution for drought monitoring application.

Both FY-3A and TERRA are polar-orbiting satellites. Channels setting of MERSI, including the center wavelength and the wave width, is basically consistent with that of MODIS, in particular the bands used for monitoring surface vegetation, atmospheric water vapor, and surface temperature. MODIS data is calibrated on orbit and it uses the complicated reprojecting technology to locate when it scans. Because of high-quality and effective monitoring, MODIS has become a widely used data source in drought monitoring. Therefore, MODIS is optimal reference data in analyzing and evaluating radiometric calibration, relative geometric location, and drought monitoring results of MERSI.



(a)



(b)

FIGURE 2: Original data (a) FY-3A MERSI, and (b) TERRA MODIS.

In order to enlarge the remote sensing data source, the objective of the present study is to evaluate the data quality and the drought monitoring capability of FY-3A MERSI data with TERRA MODIS data as a reference.

2. Materials and Methods

2.1. Study Area. The case study area is near Bohai Bay in North China (area inside the red frame, Figure 1) with latitude range from 37°N to 46°N and longitude range from 109°E to 129°E , respectively. There are a variety of land covers in the area, including vegetation, bare area, and waters.

2.2. MERSI and MODIS Data. The data of the study were extracted from MERSI on 1st October 2008, at 03:00 AM (GMT) (Figure 2(a)), and from TERRA-MODIS of the same area 15 minutes earlier (Figure 2(b)). The image of MERSI was 2000×2084 pixels, of TERRA-MODIS 1354×2030 pixels. To make it comparable with the result of MODIS data, FY-3A MERSI data was resampled to 1000-meter spatial resolution. These two kinds of data were provided by *China Meteorological Administration* (CMA) and MODIS Data Receiving Station in Wuhan University, respectively.

The orange lines in Figure 2 denote orbit central lines. Their spatial resolution has achieved the anticipated design requirement and the texture information is plentiful. The

TABLE 1: Spectral properties of MERSI (partial).

Channel	Wavelength (um)	Bandwidth (um)	Resolution (m)	$Ne\Delta\rho$ (%)	Application
3	0.650	0.05	250	0.4	Vegetation monitoring
4	0.865	0.05	250	0.45	
5	11.25	2.5	250	0.50 K	Temperature retrieve
17	0.905	0.02	1000	0.10	Atmospheric Water Vapor detection
18	0.940	0.02	1000	0.10	
19	0.980	0.02	1000	0.10	

study area is in the lower right corner of MERSI data (Figure 2(a)) and at the upper left side of MODIS data (Figure 2(b)).

2.3. Methods

2.3.1. NDVI. The Normalized Difference Vegetation Index is a satellite-derived global vegetation indicator obtained from the red, near-infrared (NIR) ratio of vegetation reflectance in the electromagnetic spectrum [15, 16]

$$NDVI = \frac{\rho_{IR} - \rho_R}{\rho_{IR} + \rho_R}. \quad (1)$$

The parameters, ρ_{IR} and ρ_R , are the reflectivity of near-infrared and red channels, respectively.

NDVI provides information on vegetation productivity and phenology over large temporal and spatial scales and has been widely used in the recent ecological studies as a proxy for vegetation productivity and phenology [17].

2.3.2. LST. The Land Surface Temperature can be calculated from the Brightness Temperature (BT) of the thermal infrared channels based on the thermal transmission equation. Because FY-3A MERSI contains only one thermal infrared channel, LST is replaced by BT of the thermal infrared channel in this paper [18].

2.3.3. TVDI. Sandholt and others extracted water stress indices (i.e., the index of temperature vegetation drought) on the basis of the simplified NDVI-LST feature space. LST_{\min} (*minimum LST*) is a straight-line paralleling with NDVI axis, and LST_{\max} (*maximum LST*) has a linear relation with NDVI. The TVDI can be obtained with formula (2) on the basis of NDVI-LST feature space [19, 20]:

$$TVDI = \frac{LST - LST_{\min}}{LST_{\max} - LST_{\min}}. \quad (2)$$

In fact, LST_{\min} in the NDVI-LST feature space varies under different vegetation coverage conditions [21–23]. Formula (2) is obtained by calculating the four Vertex Coordinates of trapezium theoretically based on the assumption that NDVI-LST feature space is trapezoidal. Therefore, LST_{\min} and LST_{\max} can be obtained through Least-squares linear fitting when NDVI-LST feature space is simplified into

a triangle. In traditional studies, the function between LST_{\min} and NDVI is defined as wet side equation while that of LST_{\max} is accordingly called dry side equation. The formulas are as follows:

$$\begin{aligned} LST_{\min} &= a_1 + b_1 NDVI, \\ LST_{\max} &= a_2 + b_2 NDVI, \end{aligned} \quad (3)$$

where a_1 and b_1 are the coefficients of the dry side equation, and a_2 and b_2 are the coefficients of the wet side equation, respectively.

In the experiment, a serial points (e.g., take LST_{\min}), $(NDVI_i, LST_{\min,i})$ is extracted, where i is the serial number of these point. Thus, the error matrix equation can be established

$$V = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} 1, NDVI_1 \\ 1, NDVI_2 \\ \vdots \\ 1, NDVI_n \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} - \begin{bmatrix} LST_{\min,1} \\ LST_{\min,2} \\ \vdots \\ LST_{\min,n} \end{bmatrix}, \quad (4)$$

where V is the error matrix, and n is the number of points.

The normal equation can be deduced from the error matrix equation based on the Least-squares algorithm

$$\begin{aligned} &\begin{bmatrix} 1, NDVI_1 \\ 1, NDVI_2 \\ \vdots \\ 1, NDVI_n \end{bmatrix}^T \begin{bmatrix} 1, NDVI_1 \\ 1, NDVI_2 \\ \vdots \\ 1, NDVI_n \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \\ &= \begin{bmatrix} 1, NDVI_1 \\ 1, NDVI_2 \\ \vdots \\ 1, NDVI_n \end{bmatrix}^T \begin{bmatrix} LST_{\min,1} \\ LST_{\min,2} \\ \vdots \\ LST_{\min,n} \end{bmatrix}, \end{aligned} \quad (5)$$

where T is the transposed operation of matrix.

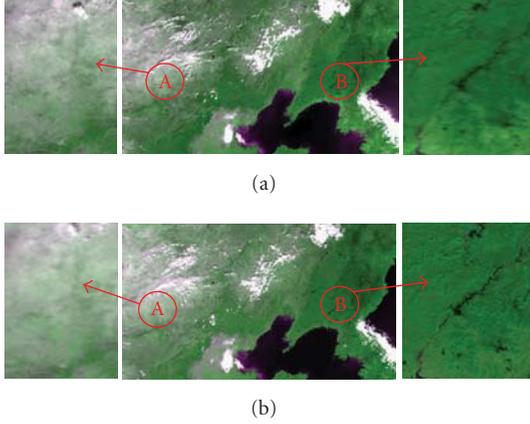


FIGURE 3: Result of radiation calibration and projection (true color: red, near-infrared, red): (a) FY-3A MERSI, and (b) TERRA MODIS.

The coefficients can be calculated based on the matrix operation:

$$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \begin{pmatrix} \begin{bmatrix} 1, \text{NDVI}_1 \\ 1, \text{NDVI}_2 \\ \vdots \\ 1, \text{NDVI}_n \end{bmatrix}^T \begin{bmatrix} 1, \text{NDVI}_1 \\ 1, \text{NDVI}_2 \\ \vdots \\ 1, \text{NDVI}_n \end{bmatrix} \\ \times \begin{bmatrix} 1, \text{NDVI}_1 \\ 1, \text{NDVI}_2 \\ \vdots \\ 1, \text{NDVI}_n \end{bmatrix}^T \begin{bmatrix} \text{LST}_{\min,1} \\ \text{LST}_{\min,2} \\ \vdots \\ \text{LST}_{\min,n} \end{bmatrix} \end{pmatrix}^{-1} \quad (6)$$

where -1 is the inversion operation of matrix.

Last, the relation square (R -square, R^2) demonstrates that this statistic measures how successful the fit is in explaining the variation of the data. A value closer to 1 indicates a better fit

$$R^2 = \frac{\left(\sum_{i=1}^n (\text{LST}_{\min,i} - \overline{\text{LST}_{\min}}) (\text{NDVI}_i - \overline{\text{NDVI}}) \right)^2}{\sum_{i=1}^n (\text{LST}_{\min,i} - \overline{\text{LST}_{\min}}) \cdot \sum_{i=1}^n (\text{NDVI}_i - \overline{\text{NDVI}})}, \quad (7)$$

where $\overline{\text{LST}_{\min}}$ is the average value of all LST_{\min} , and $\overline{\text{NDVI}}$ is the average of all NDVI.

3. Result and Discussion

After calibration, MERSI and MODIS data are projected with the latitude and longitude information recorded in scan. The relative accuracy of geometric location and radiometric calibration of the two kinds of data are also analyzed. NDVI is obtained through red and near infrared channels, and LST in the clear sky or cloud top temperature can be retrieved through thermal infrared and vapor channels. Furthermore,

it is necessary to compare and analyze the distribution character of the dry and wet sides of the two images in the NDVI-LST feature space. User's accuracy, producer's accuracy, and the overall accuracy are evaluated for TVDI with MERSI and MODIS data.

3.1. Data Preprocessing

3.1.1. Calibration. The visible channels of MERIS are directly calibrated with slope-intercept form, and revised solar altitude and Earth-Sun Distance at the imaging moment. Digital number (DN) of infrared channels is already radiation value, so the brightness temperature can be gained with Planck Formula.

The visible data of MODIS is calibrated with slope-intercept form, while the infrared channel is calibration with Planck Formula to get brightness temperature directly [24–26]. The in-lab calibration accuracy of MERSI visible and infrared channels is 7% and 1 K, respectively, but the radiation calibration accuracy of MODIS data is much higher.

Both record the latitude and longitude values in imaging; they are translated into geographic coordinates in the corresponding projection through strict transformation calculation method [27, 28]. At the same time, gray value of a pixel is extracted to a corresponding location in the new image. The new projection is Lambert Equal-angle Intercross Taper Projection (Lambert Conformal Conic). Its specific parameters include Spheroid of Krasovsky, Datum of Krasovsky, the first standard parallel latitude of 47°N , the second standard parallel latitude of 25°N , the central meridian latitude of 105°E and the origin coordinates of (5000000.0, 5000000.0) (measured in meters). The results of radiometric calibration and projection of the two data are shown in Figure 3.

It is obvious in Figure 3 that the two images have the similar color effect and the color of MERSI is slightly lighter than that of MODIS. Moreover, there is small difference in atmospheric conditions, for example, the sizes of cloud cluster are a little different at the Bohai Bay region. For region B in Figures 3(a) and 3(b), it can be easy noticed that the resolution of MERSI data, located at the right side of original images, is coarser than that of MODIS which locates at the center of the original scene. Furthermore, the partial textures information of region B in MERSI data is poorer than that of MODIS. For region A in Figures 3(a) and 3(b), it is vice versa. This is mainly due to the different positions of the study area in the original images. Some statistic parameters can be seen in Table 2, and the gray histograms of selected channels are shown in Figure 4.

The population characteristic value different of red channel reflectivity between MERSI and MODIS data is 0.05 and the population variance is 0.02. In other words, red channels of two sensors have same sensitivity to different types of ground objects. Of course, those asynchronous images and the differences of central wavelength and wave width will also have some effect on the scaling result, and this is beyond the scope of this study. The similar situation also happens to the reflectivity of near infrared channel.

TABLE 2: Compared analysis of calibration result.

Index	Red channel		Near-infrared channel		Thermal infrared channel	
	MODIS (b01)	MERSI (b03)	MODIS (b02)	MERSI (b04)	MODIS (b31)	MERSI (b05)
Wavelength	0.645	0.65	0.859	0.865	11.03	11.25
Mean	0.141	0.192	0.238	0.303	290.76	290.55
Maximum	1	1	1	1	304	301
Minimum	0.02	0.05	0.01	0.02	226	234
Medium	0.13	0.16	0.24	0.3	292	292
Std. Dev	0.101	0.126	0.12	0.152	6.868	5.643

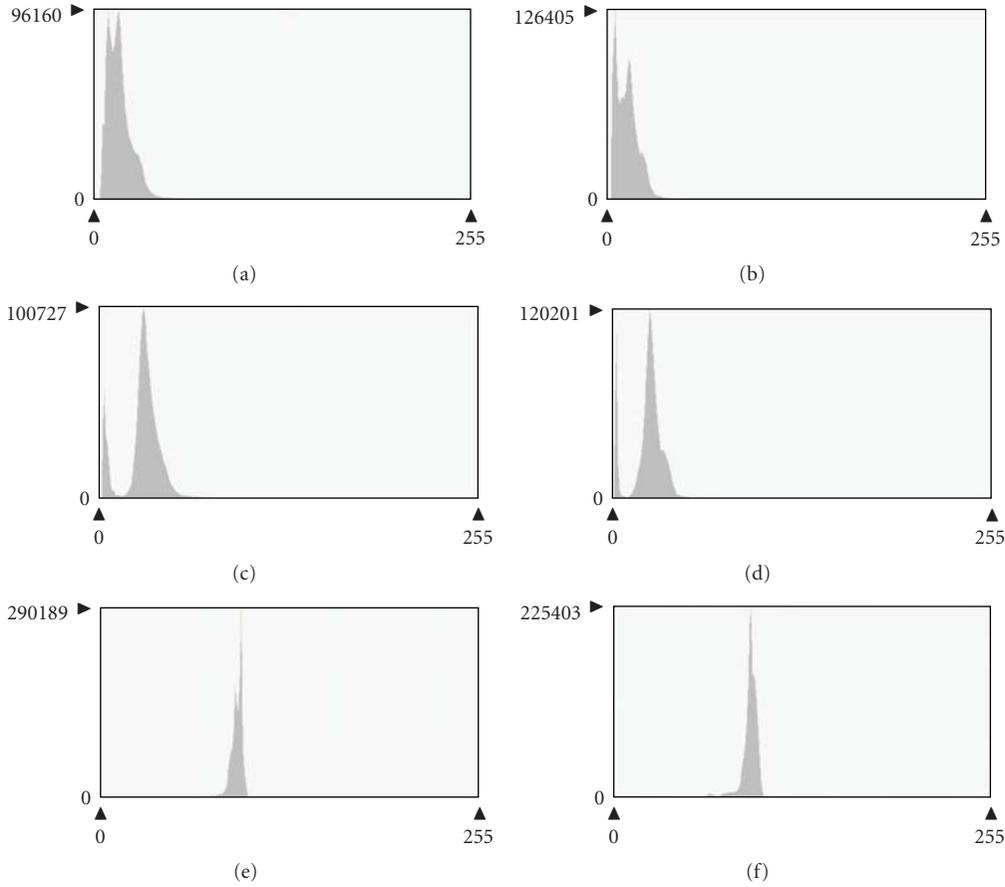


FIGURE 4: (a) MERSI red channel (reflectance * 100), (b) MODIS red channel (reflectance * 100), (c) MERSI near-infrared channel (reflectance * 100), (d) MODIS near-infrared channel (reflectance * 100), (e) MERSI thermal infrared channel (LST-200), and (f) MODIS thermal infrared channel (LST-200).

Its population characteristic value difference between the MERSI and MODIS data is 0.06 and the population variance is 0.02. As can be seen from the reflectivity histogram of near infrared channel of MERSI and MODIS data, the histogram has two-peak values. According to the reflection spectral characteristics of ground objects, the lower peak value of reflectivity mainly results from the lower reflectivity of the surface waters, as most of the study area is covered by waters. Furthermore, LST in the clear sky or cloud top temperatures retrieved from two images are of little difference, and this indicates that MERSI has a strong capability for LST retrieving.

3.1.2. Relative Geometric Location Analysis. MODIS data is of high orbital location accuracy, so it can serve as reference for analyzing geometric location accuracy of MERSI data. Fifteen control points (balanced distributed in study area) are selected manually in the study. The geometric location differences of these points are used to analyze geometry location accuracy of MERSI data. Figure 5 shows that the spatial distribution of the points and Table 3 shows the geometric location difference of each point between two images.

It is shown in Table 3 that MERSI data has quite good location accuracy when compared with MODIS data. Its

TABLE 3: Geometric location accuracy of MERSI data (unit: meter).

Point no.	MODISX	MODISY	MERSIX	MERSIY	X_Diff	Y_Diff	RMS
1	5645738	5524095	5645718	5524098	-682	186	707.32
2	5932799	5861170	5932795	5862104	-552	-317	637.26
3	6491754	5291080	6491693	5292060	-104	-176	205.46
4	6854745	5403135	6853753	5404095	-68	121	139.76
5	7000293	5764656	6998805	5765990	-160	128	205.96
6	6636750	6044083	6635740	6045110	-90	353	364.98
7	6371813	5412078	6371707	5413082	241	-173	297.46
8	6129717	6003122	6128731	6003088	202	705	734.52
9	5576752	6133008	5573799	6133098	233	15	233.74
10	5417760	5347094	5415739	5347072	504	-136	522.30
11	6884743	6095120	6882747	6097091	67	-381	387.63
12	6299718	5735132	6298756	5736088	986	11	986.91
13	6338741	6141127	6337732	6142106	-143	137	198.29
14	5409658	5642143	5406747	5642105	612	-46	614.34
15	5798758	5283109	5798707	5283129	-229	232	326.44
Total difference			X: 415.0335	Y: 213.1443	TOTAL: 466.5654		

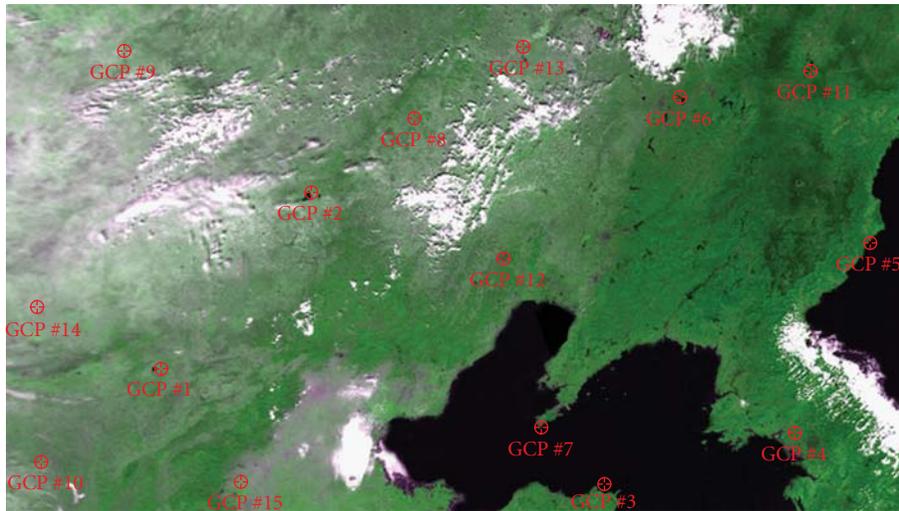


FIGURE 5: The distribution of the matching points.

overall difference is 0.4666 pixels, with X direction difference of 0.415 pixels and Y direction difference of 0.2131 pixels. The maximum difference point for X is the point no. 12 with a small difference of Y , mainly because of manual point-choosing. Actual spatial resolution of the points at the side is coarser than those near nadir, so the points at the left side of MODIS (no. 1, no. 2, no. 8, no. 12, and no. 14) have poor matching with their corresponding points near nadir of MERSI.

Based on the analysis of calibration results, MERSI can retrieve similar reflectivity, LST, and cloud top temperature as MODIS does. The geometric location difference between MERSI and MODIS is less than half pixel, while difference of some areas reaches one pixel, but it is still within the tolerance. In other words, the geolocation of MERSI is acceptable. Therefore, the MERSI data has potential for

drought monitoring on basis of calibration and relative geometric location analysis.

3.2. Cloud Detection. NDVI and LST, the two key parameters for TVDI method, cannot be retrieved when remote sensing data is covered by clouds, so TVDI is inapplicable for the cloud-covered areas. Consequently, to identify whether an image is qualified for drought monitoring, cloud detection should be performed before deducing parameters [29]. In addition, areas covered by water need to be identified and ignored in case of error in establishing NDVI-LST feature space.

Cloud in visible band shows high reflectance, and remote sensing image with 0.66 μm is ideal for distinguishing borders between land and cloud [30]. The spectrum feature of cloud in NIR band (near 0.936 μm) is mainly relevant

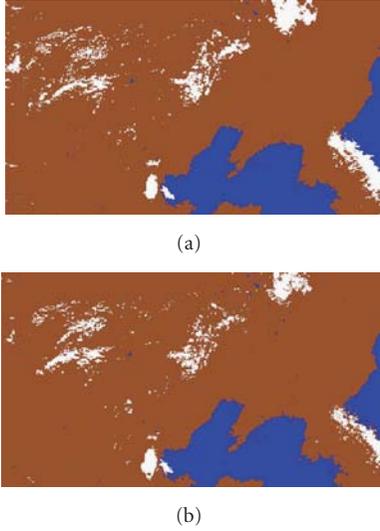


FIGURE 6: Result of cloud detection: (a) FY-3A MERSI, and (b) TERRA MODIS (brown is the clear sky, blue is water, white is cloud).

to water vapor content in atmosphere, so it can help to indicate the moisture feature, namely, absorption window. Because 0.66 μm and 0.936 μm clouds form sharp contrast with spectrum features of all kinds of ground objects, cloud information can be obtained by normalization, which not only gives prominence to cloud information and partially eliminate ill effects caused by solar altitude, satellite sweep angle and atmosphere route radiation. *Normalized Cloud Detection Index* (CDI) [12, 21] is expressed as

$$\text{CDI} = \frac{\rho_{0.66\mu\text{m}} - \rho_{0.936\mu\text{m}}}{\rho_{0.66\mu\text{m}} + \rho_{0.936\mu\text{m}}}. \quad (8)$$

As cloud reflection spectrum has reflection characteristics at 0.66 μm and its absorption window at 0.936 μm is influenced by water vapor, the CDI is positive ($\text{CDI} > 0$), because soil reflection spectra has little difference in reflection properties between 0.66 μm and 0.936 μm , the CDI is close to 0; vegetation reflection spectra has low reflection at 0.66 μm and high reflection at 0.936 μm , the CDI is negative ($\text{CDI} < 0$).

As the reflectance of water is higher than vegetation near the red band and the reflectance of vegetation is obviously higher than water round the NIR band (0.841 μm –0.876 μm) [31, 32], a typical bimodal distribution appears in the histogram in Figures 4(c) and 4(d). Thus, NDVI is adopted to enhance land-water contrast.

Based on the cloud detection and water extraction algorithms discussed above, it can be concluded that MERSI and MODIS data can have cloud and water detection with the same algorithm and the results is shown in Figure 6.

Figure 6 shows that the results of MERSI and MODIS are unanimous in most areas. However, as a result of difference in imaging time, there is a slight difference in the cloud detection at the top left corner of images due to small atmospheric condition change.

3.3. Extraction of Drought Feature Parameters. For MERSI and MODIS, NDVI is usually calculated by red band and NIR band. Meanwhile, MERSI has only one TIR band (10~12.5 μm with 250 m resolution in nadir point), which can be used to extract LST in the clear sky or cloud top temperature. Qin Zhihao deduced a simple, feasible LST algorithm with high accuracy according to the equation of land surface heat conduction [33, 34] and this method can be quoted to process MERSI data.

From previous studies, it is known that the distribution of the wet and dry sides is significant in the two-dimensional feature space composed by NDVI and LST from MERSI. It is necessary to analyze the characteristics of the NDVI-LST feature space and determine the wet and dry sides.

LST is calculated with the NDVI datasets and single-window algorithm, LST_{max} , LST_{min} , of different NDVI conditions is extracted on the basis of 0.01 NDVI step (NDVI is among 0.3 to 0.7 for medium vegetation coverage), and then NDVI-LST feature space is obtained (Figure 7).

It is shown in Figure 7 that LST_{max} and LST_{min} have better correlation with NDVI in MERSI than in MODIS, but their linear correlations are quite consistent. This may be due to the difference of cloud detection, that is to say, the cloud top temperature may be considered as the minimum LST in the clear sky in the process of selecting MODIS LST samples. From Figure 7, MODIS has smaller LST_{min} and higher LST_{max} for the same NDVI value. This demonstrates that MODIS data is doing better in retrieving LST in the clear sky than MERSI. Based on the principle of least squares solver fitting, the relationship functions are regressed in the NDVI-LST feature space. Because wet and dry side fitting of the low and high vegetation coverage (with NDVI ranging from 0 to 0.30 and from 0.70 to 1, resp.) is quite instable, only middle NDVI of 0.30 to 0.70 is used in regression.

The fitting equations of wet and dry sides based on MERSI are as follows.

Dry side equation:

$$\text{LST}_{\text{max}} = 34.550 - 17.90 \text{NDVI}, \quad R^2 = 0.8307. \quad (9)$$

Wet side equation:

$$\text{LST}_{\text{min}} = -1.643 + 28.47 \text{NDVI}, \quad R^2 = 0.9541. \quad (10)$$

The above formulas demonstrate that if the slope of dry side is less than zero, LST_{max} will decrease with the increase of vegetation coverage. Oppositely, the minimum LST is vice versa.

The fitting R -squares of MERSI wet and dry sides are 0.8307 and 0.9541, respectively. It shows that changes of LST_{max} and LST_{min} are quite consistent with the changes of NDVI, and the mechanism of drought is well demonstrated. The fitting R -squares of MODIS are 0.6852 and 0.6904, respectively.

3.4. Drought Monitoring. It is found through the above analysis that the wet and dry sides possess high-fitting relevance in the NDVI-LST feature space of MERSI and it means that MERSI is suitable for drought monitoring. TVDI

TABLE 4: TVDI confusion matrices between MERSI and MODIS.

TERRA	FY-3A						Producer accuracy
	High	Middle	Low	Normal	Wet	Cloud	
High	0.1281	0.1949	0.056	0.0287	0.1276	0.097	20.235
Middle	0.105	8.0814	7.7763	0.4183	0.3291	0.602	46.6786
Low	0.1833	1.3909	18.945	4.3751	1.1041	0.533	71.4042
Normal	0.110	0.1072	2.3245	7.5212	2.4723	0.239	58.8742
Wet	0.2405	0.0579	0.3669	2.3293	21.2561	0.266	86.6994
Cloud	0.204	0.4602	0.5915	0.31	0.4686	16.195	88.8404
User accuracy	13.1908	78.517	63.024	50.199	82.523	90.296	Correct: 72.1274 Acceptable: 93.8306

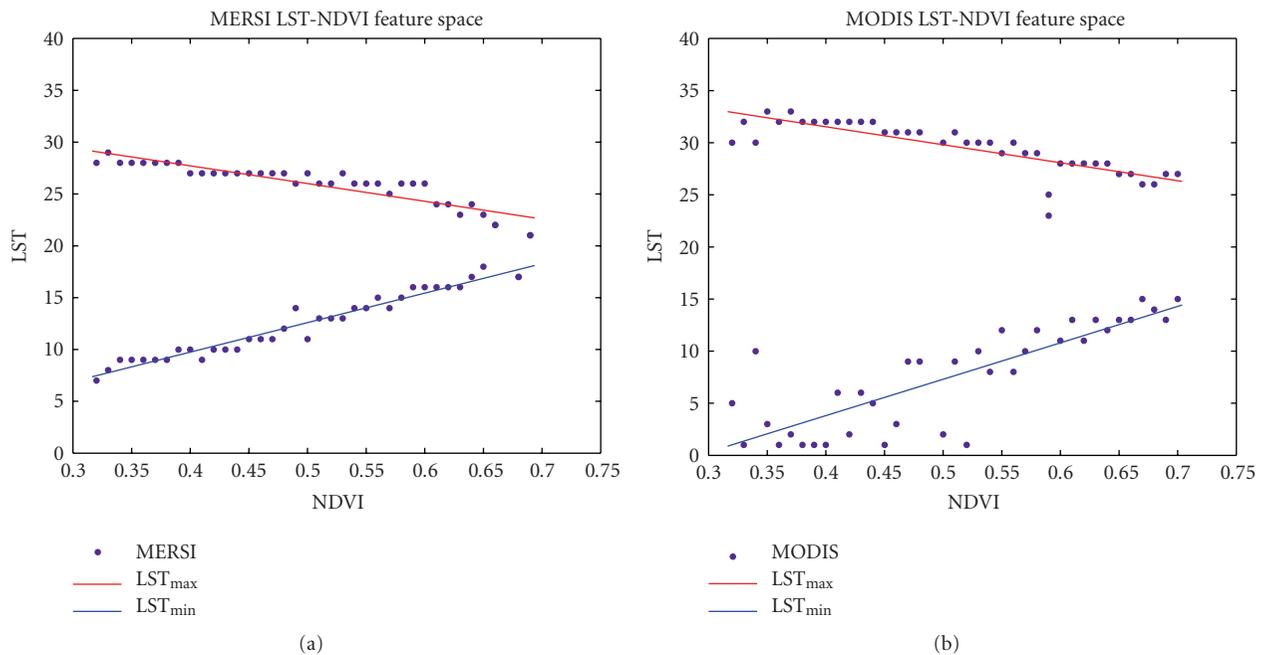


FIGURE 7: The wet and dry sides in feature space: (a) FY-3A MERSI, and (b) TERRA MODIS.

is used to analyze the drought monitoring capability of FY-3A MERSI in the study. MODIS has been proved to be one of the most successful data for drought monitoring and is naturally utilized as reference data. The monitoring results and comparison of MERSI and MODIS are as shown in Figure 8.

Table 4 is comparison of TVDI confusion matrices between FY-3A MERSI and TERRA MODIS. User's accuracy, producer's accuracy, and overall accuracy of MERSI are analyzed with MODIS TVDI as a reference. The overall accuracy contains the correct and acceptable accuracy, where the correct accuracy indicates that the monitoring results of MERSI TVDI and MODIS TVDI are unanimous. The definition of the acceptable accuracy is that the difference of grade results is lower than one level. For instance, it is acceptable if one pixel is shown as severe drought in MERSI TVDI and its corresponding point shown as medium drought in MODIS TVDI. It is clear in Table 4 that MERSI has almost the same drought monitoring result with MODIS.

In other words, absolutely correct accuracy is 72.1274 and the acceptable accuracy is as high as 93.8306. Thus, MERSI is as strong as MODIS in drought monitoring.

4. Conclusion

The following conclusions can be reached on the basis of above comparison and analysis.

- (1) FY-3A MERSI data enjoys a high quality. The relative difference of calibration results between MERSI and MODIS is negligible, and temperature retrieving capability of MERSI is as good as MODIS. In addition, it is also proved that their cloud detection results have strong consistency. Furthermore, the overall difference of geo-location between MERSI and MODIS is about 0.4666 pixels, in other words, MERSI data possess wonderful geo-location capability.

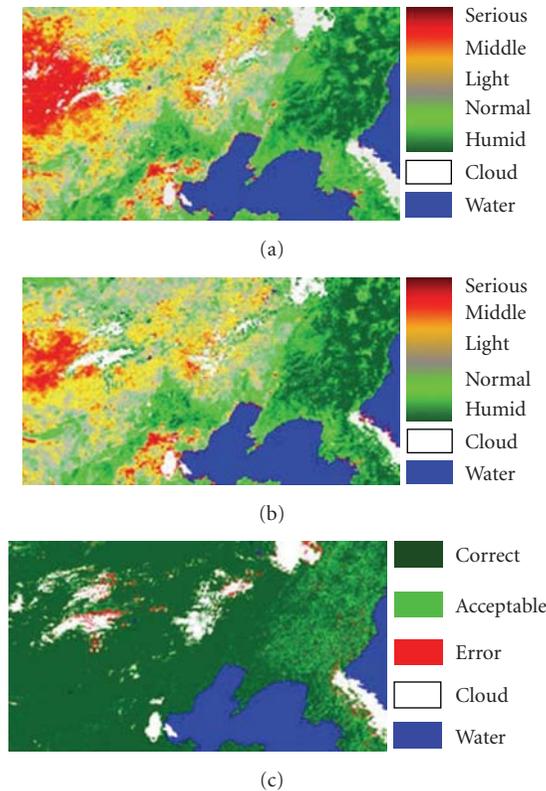


FIGURE 8: (a) TVDI from FY-3A MERSI, (b) TVDI from MODIS TERRA, and (c) TVDI Difference between MERSI and MODIS.

- (2) The LST_{max} decreases with the increase of vegetation coverage, vice versa for LST_{min} . The fitting R-squares of wet and dry sides extracted from MERSI are as high as 0.8307 and 0.9541, respectively. It means that changes of LST_{max} and LST_{min} have high consistency with those of NDVI, and the mechanism of drought is well presented. While, the fitting R-squares of MODIS data (for wet and dry side) are, respectively, 0.6852 and 0.6904 which are lower than that of MERSI. From the NDVI-LST feature space of MERSI, the scatter points' distribution of wet side is highly correlated with that of dry side, and this is a clear demonstration of drought information.
- (3) The TVDI confusion matrices indicate that FY-3A MERSI possesses the similar capability to MODIS, with absolute accuracy of 72.1274 and acceptable accuracy of 93.8306. Therefore, it is possible for us to obtain the drought products in operation with 250 m spatial resolution and global scale with MERSI as a new kind of data.

Acknowledgments

The authors are grateful to the members of the MODIS Data Receiving Station, Wuhan University, China. This paper is granted by FY-3A Satellite Application in Drought Monitoring and Early Warning in Northwest China which

is subtask of FY-3A Satellite Application Projection from China Meteorological Administration (CMA). Meanwhile, this paper is also granted the AMD University Plan.

References

- [1] D. A. Wilhite and M. H. Glantz, "Understanding the drought phenomenon: the role of definitions," *Water International*, vol. 10, no. 3, pp. 111–120, 1985.
- [2] R. Nemani, L. Pierce, S. Running, and S. Goward, "Developing satellite-derived estimates of surface moisture status," *Journal of Applied Meteorology*, vol. 32, no. 3, pp. 548–557, 1993.
- [3] A. Volcani, A. Karnieli, and T. Svoray, "The use of remote sensing and GIS for spatio-temporal analysis of the physiological state of a semi-arid forest with respect to drought years," *Forest Ecology and Management*, vol. 215, no. 1–3, pp. 239–250, 2005.
- [4] T. Tadesse, J. F. Brown, and M. J. Hayes, "A new approach for predicting drought-related vegetation stress: integrating satellite, climate, and biophysical data over the U.S. central plains," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 4, pp. 244–253, 2005.
- [5] P. R. Bajgirani, A. A. Darvishsefat, A. Khalili, and M. F. Makhdom, "Using AVHRR-based vegetation indices for drought monitoring in the Northwest of Iran," *Journal of Arid Environments*, vol. 72, no. 6, pp. 1086–1096, 2008.
- [6] A. T. Jeyaseelan and F. N. Kogan, "Evaluation of GVI based indices for drought early warning in India," in *Disaster Forewarning Diagnostic Methods and Management*, vol. 6412 of *Proceedings of SPIE*, pp. 4120–4120, 2006.
- [7] G. Marshall and X. Zhou, "Drought detection in semi-arid regions using remote sensing of vegetation indices and drought indices," in *Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS '04)*, vol. 3, pp. 1555–1558, September 2004.
- [8] L. Liu, D. Xiang, X. Dong, and Z. Zhou, "Improvement of the drought monitoring model based on the cloud parameters method and remote sensing data," in *Proceedings of the 1st International Workshop on Knowledge Discovery and Data Mining (WKDD '08)*, pp. 293–296, 2008.
- [9] W. Mo, Z. Wang, H. Sun, L. Ma, and L. He, "Remote Sensing Monitoring of Farmland Drought Based on Vegetation Supply Water Index," *Journal of Nanjing Institute of Meteorology*, vol. 29, no. 3, pp. 396–401, 2006.
- [10] C. Naira, L. Robert, M. Ramata, and T. Marouane, "Surface soil moisture status over the Mackenzie River Basin using a temperature/vegetation index," in *Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS '07)*, pp. 1846–1848, 2007.
- [11] N. R. Patel, R. Anapashsha, S. Kumar, S. K. Saha, and V. K. Dadhwal, "Assessing potential of MODIS derived temperature/vegetation condition index (TVDI) to infer soil moisture status," *International Journal of Remote Sensing*, vol. 30, no. 1, pp. 23–39, 2008.
- [12] W. Li, S. Y. Mao, and W. Chen, "A new method of cloud detection in MODIS image," in *Proceedings of the 7th International Conference on Electronic Measurement and Instruments*, vol. 7, pp. 281–285, 2005.
- [13] G. Cai, *MODIS data based thermal inertia and land surface temperature modeling and their applications in determination of soil moisture and heat exchange*, doctoral dissertation, M.S. thesis, Institute of remote sensing applications Chinese academy of sciences, 2006.

- [14] L. Liu, D. Xiang, Z. Zhou, and X. Dong, "Analyses the modification functions of the drought monitoring model based on the cloud parameters method," in *Proceedings of the 1st International Congress on Image and Signal Processing (CISP '08)*, vol. 4, pp. 687–691, 2008.
- [15] B. C. Reed, J. F. Brown, D. VanderZee, T. R. Loveland, J. W. Merchant, and D. O. Ohlen, "Measuring phenological variability from satellite imagery," *Journal of Vegetation Science*, vol. 5, no. 5, pp. 703–714, 1994.
- [16] S. Hamel, M. Garel, M. Festa-Bianchet, J.-M. Gaillard, and S. D. Côté, "Spring Normalized Difference Vegetation Index (NDVI) predicts annual variation in timing of peak faecal crude protein in mountain ungulates," *Journal of Applied Ecology*, vol. 46, no. 3, pp. 582–589, 2009.
- [17] J. T. Kerr and M. Ostrovsky, "From space to species: ecological applications for remote sensing," *Trends in Ecology & Evolution*, vol. 18, no. 6, pp. 299–305, 2003.
- [18] Y. Julien and J. A. Sobrino, "The Yearly Land Cover Dynamics (YLCD) method: an analysis of global vegetation from NDVI and LST parameters," *Remote Sensing of Environment*, vol. 113, no. 2, pp. 329–334, 2009.
- [19] I. Sandholt, K. Rasmussen, and J. Andersen, "A simple interpretation of the surface temperature/vegetation index space for assessment of surface moisture status," *Remote Sensing of Environment*, vol. 79, no. 2-3, pp. 213–224, 2002.
- [20] S. Qi, C. Wang, and Z. Niu, "Evaluating soil moisture states in China using the temperature/vegetation dryness index(TVDI)," *Journal of Remote Sensing*, vol. 15, pp. 420–427, 2003.
- [21] W. Li and D. Li, "The universal cloud detection algorithm of MODIS data," in *Geoinformatics 2006: Remotely Sensed Data and Information*, vol. 6419 of *Proceedings of SPIE*, p. 4190, October 2006.
- [22] C. Li and H. Li, "Study on winter wheat drought monitoring by TVDI in Hebei Province," in *Remote Sensing and Modeling of Ecosystems for Sustainability III*, vol. 6298 of *Proceedings of SPIE*, pp. 2981–2981, San Diego, Calif, USA, August 2006.
- [23] C. Jin, Q. M. Qin, L. Zhu, P. Nan, and A. Ghulam, "TVDI based crop yield prediction model for stressed surfaces—a case study on Ningxia Huizu autonomous region of China," in *Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS '08)*, pp. 4656–4658, 2008.
- [24] J. Tong, K. Qiu, X. Li, and Z. Yang, "Radiometric cross-calibration of the FY1D/VIRR and EOS/MODIS in the visible and near-infrared spectral bands," *Journal of Remote Sensing*, vol. 4, pp. 349–356, 2005.
- [25] G.-K. Zhang, C. Chen, F. Xing, H.-Y. Zhang, and Y.-S. Zhao, "Spectral radiometric calibration research of Quick Bird digital image," *Spectroscopy and Spectral Analysis*, vol. 28, no. 3, pp. 494–498, 2008.
- [26] Z. Zhang and G. He, "Radiometric calibration of Landsat 5 TM data," *Science and Technology Review*, vol. 7, pp. 54–58, 2008.
- [27] Z. M. Cheng, L. Lu, and X. M. Qiu, "Make use of the space projection method to carry on the geometric exact correction to the remote sensing picture," *Science Mosaic*, vol. 10, pp. 103–105, 2005.
- [28] F. Duan, L. Zhu, and L. Gao, "Realization and optimization of multi-projection transformation on remote sensing image," *Journal of Image and Graphics*, vol. 5, pp. 608–610, 2005.
- [29] D. C. Bamber, S. Mackin, and P. Palmer, "Cloud detection and height estimation through registration of disaster monitoring constellation imagery," in *Remote Sensing of Clouds and the Atmosphere XI*, vol. 6362 of *Proceedings of SPIE*, pp. 448–457, September 2006.
- [30] T. Cheng, X. Gu, L. Chen, T. Yu, and G. Tian, "Cloud detection based on the spectral, multi-angular, and polarized characteristics of cloud," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '08)*, pp. 3321–3324, Barcelona, Spain, June 2008.
- [31] X. Wu and Q. Cheng, "Study on methods of cloud identification and data recovery for MODIS data," in *Remote Sensing of Clouds and the Atmosphere XII*, vol. 6745 of *Proceedings of SPIE*, pp. 7450–7450, Florence, France, September 2008.
- [32] H. Wang, Y. He, and H. Guan, "Application of support vector machines in cloud detection using EOS/MODIS," in *Remote Sensing Applications for Aviation Weather Hazard Detection and Decision Support*, vol. 7088 of *Proceedings of SPIE*, p. 880, San Diego, Calif, USA, August 2008.
- [33] Z.-H. Qin, M.-H. Zhang, A. Karnieli, and P. Berliner, "Monowindow algorithm for retrieving land surface temperature from Landsat TM6 data," *Acta Geographica Sinica*, vol. 56, no. 4, pp. 456–466, 2001.
- [34] D. M. Song, N. K. Hu, Y. L. Wu, and X. Y. Li, "The study on the ground surface parameters in Minqin Oasis in Gansu Province with using remote sensing data," in *Proceeding of the International Conference on Information Technology and Environmental System Science*, vol. 4, pp. 339–343, 2008.

Research Article

Multibandwidth Kernel-Based Object Tracking

Aras Dargazany, Ali Soleimani, and Alireza Ahmadyfard

Department of Electrical and Robotics Engineering, Shahrood University of Technology, Shahrood, Iran

Correspondence should be addressed to Aras Dargazany, aras.dargazany@gmail.com

Received 1 August 2009; Revised 2 December 2009; Accepted 20 April 2010

Academic Editor: Ce Zhu

Copyright © 2010 Aras Dargazany 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.

Object tracking using Mean Shift (MS) has been attracting considerable attention recently. In this paper, we try to deal with one of its shortcomings. Mean shift is designed to find local maxima for tracking objects. Therefore, in large target movement between two consecutive frames, the local and global modes are not the same as previous frames so that Mean Shift tracker may fail in tracking the desired object via localizing the global mode. To overcome this problem, a multibandwidth procedure is proposed to help conventional MS tracker reach the global mode of the density function using any starting points. This gradually smoothing procedure is called Multi Bandwidth Mean Shift (MBMS) which in fact smoothens the Kernel Function through a multiple kernel-based sampling procedure automatically. Since it is important for us to have less computational complexity for real-time applications, we try to decrease the number of iterations to reach the global mode. Based on our results, this proposed version of MS enables us to track an object with the same initial point much faster than conventional MS tracker.

1. Introduction

Using a kernel function as a density estimator are methods in image processing which drew much attention. Object tracking [1–7] with MS is a nonparametric technique, introduced in [8–10]. Actually, it is a mode detector algorithm in the density distribution space. This method assigns weights to the pixels within a window. It is proposed that MS iteration step size can be adapted [9]. It also said MS and Newton algorithm are connected [11]. MS is utilized in object tracking as an alternative for particle filtering tracking [2, 12, 13]. The Kernel scale can also be updated [3]. To avoid changes in target representation, multiple kernel functions can be used in order to maintain the pixel location values [5, 7, 14]. Another way toward using a kernel function is similarity function [15]. The main problems with MS: The first is failure caused in rapid movement of an object in two consecutive frames; the second is that MS is a local optimizing technique, that is, we should not expect to localize the global and optimal mode; the third is slow tracking. In fact, MS searching procedure is initialized by the tracked object in the last frame. Flattening the optimization surface (created by windowing around the initial point) is used to handle

sampling method through finding the global maximum [16–18]. The multi-bandwidth kernel function can flatten the cost function (i.e., Likelihood Surface) to avoid local modes rather than global one. Through increasing the bandwidth of the kernel, all modes will be unified into one main mode whose peak location is the nearest point to the global optimal mode as we desired. Then, by using the convergence point of this mode of unified likelihood surface, the first step will be easily modified through MS iterations. The bandwidth determines the degree of flatness of the surface. In this way, the number and the position of modes are being evolved and changed slowly. Lower computational cost is more desirable in real-time tracking. Using nearest neighbor of a sample in MS iteration and kd-tree to reduce the number of nearest neighbors decrease the cost [1, 19]. The density distribution is described through clustering the feature space also to lessen the cost [20]. Quasi-Newton is also then put forward to linearize the rate of convergence [11, 21]. Many attempts have been made to enhance the efficient bound optimization algorithm [22–24]. Our proposed method can detect an object globally. This technique can prevail over the recent methods and also extend the last technical efforts [25]. This paper is summarized as follows: an overview and reviewing

analysis over MS is given in Section 2. The proposed Method will be elaborated along with enhancement of MS iterations in detecting the global mode of cost function in Section 3. Implementation of the proposed method and how faster this proposed method can function are revealed in Section 4. Using color as feature, automatic initialization, how to cope with background problem, and being a self-localizer are all stated with experimental results in Section 5. At the end, the conclusion comes in Section 6.

2. Mean Shift

The conventional and original version is a nonparametric kernel density estimator

$$\hat{f}_K(x) = \frac{1}{n} \sum_{i=1}^n K_{\mathbf{H}}(x - x_i). \quad (1)$$

According to (1), $K_{\mathbf{H}}(x) = |\mathbf{H}|^{-(1/2)} K(\mathbf{H}^{-(1/2)}x)$ where K is a kernel function (i.e., here Gaussian function) with 2D bandwidth as $\mathbf{H} \in R^{d \times d}$, which is equal to $\mathbf{H} = h^2 \mathbf{I}$ (i.e., identity matrix) [9, 10]. In this way, we can have kernel estimator as

$$\hat{f}_K(x) = \frac{c_k}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{x - x_i}{h}\right\|^2\right), \quad (2)$$

formulated through the following equation:

$$\begin{aligned} \hat{\nabla} f_K(x) &\equiv \nabla \hat{f}_K(x) = \frac{2c_k}{h^2 c_g} \hat{f}_G(x) \cdot m_G(x) = 0, \\ \hat{f}_G(x) &= \frac{c_g}{nh^d} \sum_{i=1}^n g\left(\left\|\frac{x - x_i}{h}\right\|^2\right), \end{aligned} \quad (3)$$

$$m_G(x) = \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)} - x,$$

$K(x)$ is kernel function and its gradient is $g(x)$. Here $m_G(x)$ is a vector created by MS procedure. These equations result in:

$$x \leftarrow \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}. \quad (4)$$

In this paper, it is proposed to use multiple bandwidths in a conventional mean shift tracker (i.e., a broad bandwidth tracking a larger motion). A broad bandwidth played the central role to help tracking a larger motion. Due to the smoothness incorporated by the large bandwidth, the fixed point iteration can track due to converging faster. It is argued that the bandwidths can be automatically obtained. However, it can be seen below the overall algorithm for choosing effect of the algorithm in detail. The algorithm is also shown and explained below. Since we are dealing with an automatic bandwidth selection, the optimal bandwidth is the main and final goal after smoothing likelihood surface (i.e., cost function). The optimal bandwidth is the bandwidth in which

global mode and other local modes are clear enough to seek (i.e., initial state of likelihood surface) and in the same bandwidth seeking about the global mode will take place by MBMS (Multi-Bandwidth Mean Shift).

In fact, there are a variety of ways to select the optimal bandwidth for an automatic optimal bandwidth selecting procedure but in this automatic procedure, first we calculate the optimal bandwidth according to [26, 27] in order to have a minimum AMISE (i.e., Asymptotic Mean Integrated Square Error) as introduced in [26, 27], AMISE is an estimate of distance between two different densities for evaluating the performance of a kernel density estimator [28]. Having found the optimal bandwidth, we mean to find a large bandwidth sufficiently enough through the proposed MB (i.e., Multi-Bandwidth) procedure. By sufficiently this large bandwidth, we desire a bandwidth which is much larger than the optimal bandwidth with the minimum AMISE.

This large bandwidth is meant to create a unimodal likelihood surface which is consisting of a single mode. This mode will be used by Mean Shift procedure in the first run. It does not matter whether this mode is global or not. Having found the final location of this mode using Mean Shift helps us achieve finally the global mode via the MB procedure in the optimal bandwidth (i.e., very last selected bandwidth) using MS iterations.

Having observed the above equations, therefore we can have now the optimal bandwidth at the last stage of MBMS, in which all modes especially global mode are clear. If we use three or four times as much as the optimal bandwidth as you can see in 1D and 2D likelihood surfaces (i.e., Cost Function) Figures 1 and 2, we will have a unimodal surface then. By detecting this lonely single mode, the closest point in the best basin of attraction, we can reach the real global mode in more stable way without facing any other local mode.

3. The Proposed Method

Monotonic decrease of bandwidth ends to h_0 as the experimental optimal bandwidth. Figure 1 shows a one-dimensional Gaussian mixture with some modes which are all going to be unified and to change into a unimodal surface via the bandwidth increasing trend [29]. With a multi-bandwidth process, it will lead to the evolution of modes as illustrated in Figure 1. In this way, seeking the global mode would not be trapped in the local modes. The algorithm and the procedure are very simple as you see in the Figure 1. The great advantage of this method is that the starting point location is not important at all. This paper is based specifically on Gaussian kernel due to its monotonicity [30]. The optimal bandwidth is h_0 . The Gaussian kernel can also be transformed to reduce the cost [29, 31, 32].

There is also a method to find the minima instead of maxima [33]. In order to find an important mode among others, variable selection of bandwidth is utilized [34]. In our work, we can guarantee the tracking with the proper and manual bandwidth selection but it also can be extended through automatic estimation of bandwidth using some features like the target region variance [4]. At the largest

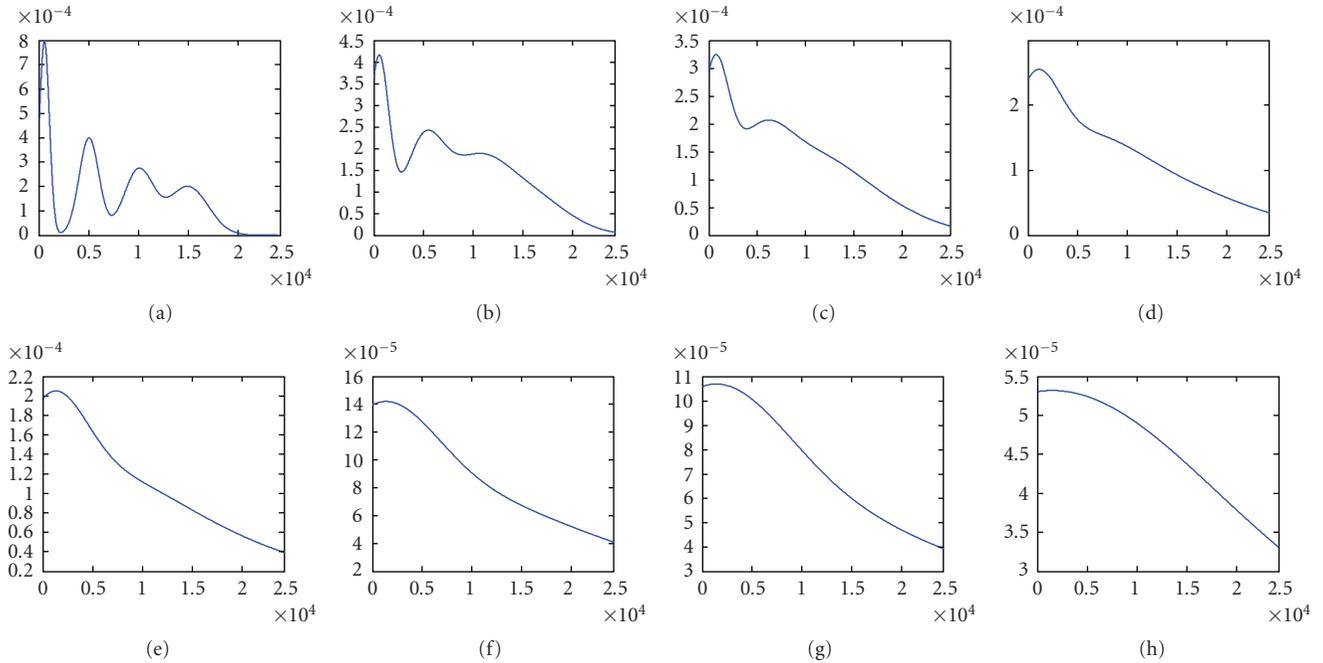


FIGURE 1: One-dimensional Gaussian mixture surface changes with the bandwidth: 500, 1000, 1500, 2500, 3500, 5500, 7500, and 15500.

bandwidth, we have a unimodal surface. Via MS iteration the convergence at the largest bandwidth (unimodal) will be easily achieved as it illustrated in Figure 2. It is mainly argued that there are multiple bandwidths in the proposed method. However, it is shown how these multiple bandwidths collaborate during bandwidth selection procedure in MBMS (i.e., Multi-Bandwidth Mean Shift). Here, the strategy of the use of these multiple bandwidth selection is given below.

MBMS algorithm:

- (1) selecting the sequence of bandwidth h_m as $m = M, \dots, 0$ (i.e., Multi-bandwidth smoothing procedure),
- (2) a starting location for first MB (multi-bandwidth) procedure and converge using h_m in $\hat{f}_{h_m, K}(\cdot)$ which is $\hat{x}^{(M)}$ using MS (Mean Shift),
- (3) we run MS for each h_m (i.e., indicated in 1) to get the convergence position $\hat{x}^{(m)}$ with the initial position $\hat{x}^{(m+1)}$, this means that the convergence position achieved in previous bandwidth. Finally, reaching $\hat{x}^{(0)}$ means that we have located the global mode on the likelihood surface (i.e., Cost Function).

4. The Improved MS

Multi-bandwidth selecting and utilizing cost us some delay and computational burden. MS is proved to be a quadratic bound optimization according to [6, 9, 11]. This algorithm is an ascending approach with adaptable iteration size. In MS, no default parameter is needed to be set at first. It has been reasoned that a Gaussian mixture model can be optimized by fixed-point bound optimization method [35]. This method

is also applicable to the surface created by other kernel functions [24]. Because of its speed, the bound optimization algorithms has a slow convergence [22–24]. Suppose that the similarity function for two frames is defined with $\rho(x, x^{(\kappa)})$ so that the adaptable step size can be determined as

$$x^{(\kappa+1)} = x^{(\kappa)} + \beta \left[\arg \max_x \rho(x, x^{(\kappa)}) - x^{(\kappa)} \right]. \quad (5)$$

β is a learning rate as you see above. If we want a more reliable convergence, it should be $0 < \beta < 2$ as proved and generalized in [22, 24]. In our work, it is set $\beta = 1$, then it begins initializing and running MS iteratively. Gaussian kernel function is helping us because its gradient equals to the original function so that their application in weight assignment and flattening the surface with the same bandwidth is quite similar [28]. This analysis is validated by our experimental results.

4.1. Experimental Result. We run the two algorithms for the same data sets and insert the results in Table 1 and Figures 4 and 5. The starting point is initialized in different location. As you see the improved MS achieves better results than original MS and we can claim that it functions even faster than PF [30, 36]. The proposed algorithm outperforms MS in number and step-size of iterations. It can be also compared with the Quasi-Newton method implemented in [21, 37–39]. This developed MS is implemented in MATLAB programming space and is run in an Asus notebook inside Vista windows Pentium 2 GHz CPU. The proposed MS is more different than conventional MS. In comparison with other linear bound fixed-point optimization algorithms such as Quasi-Newton [21], our proposed method is more accurate and

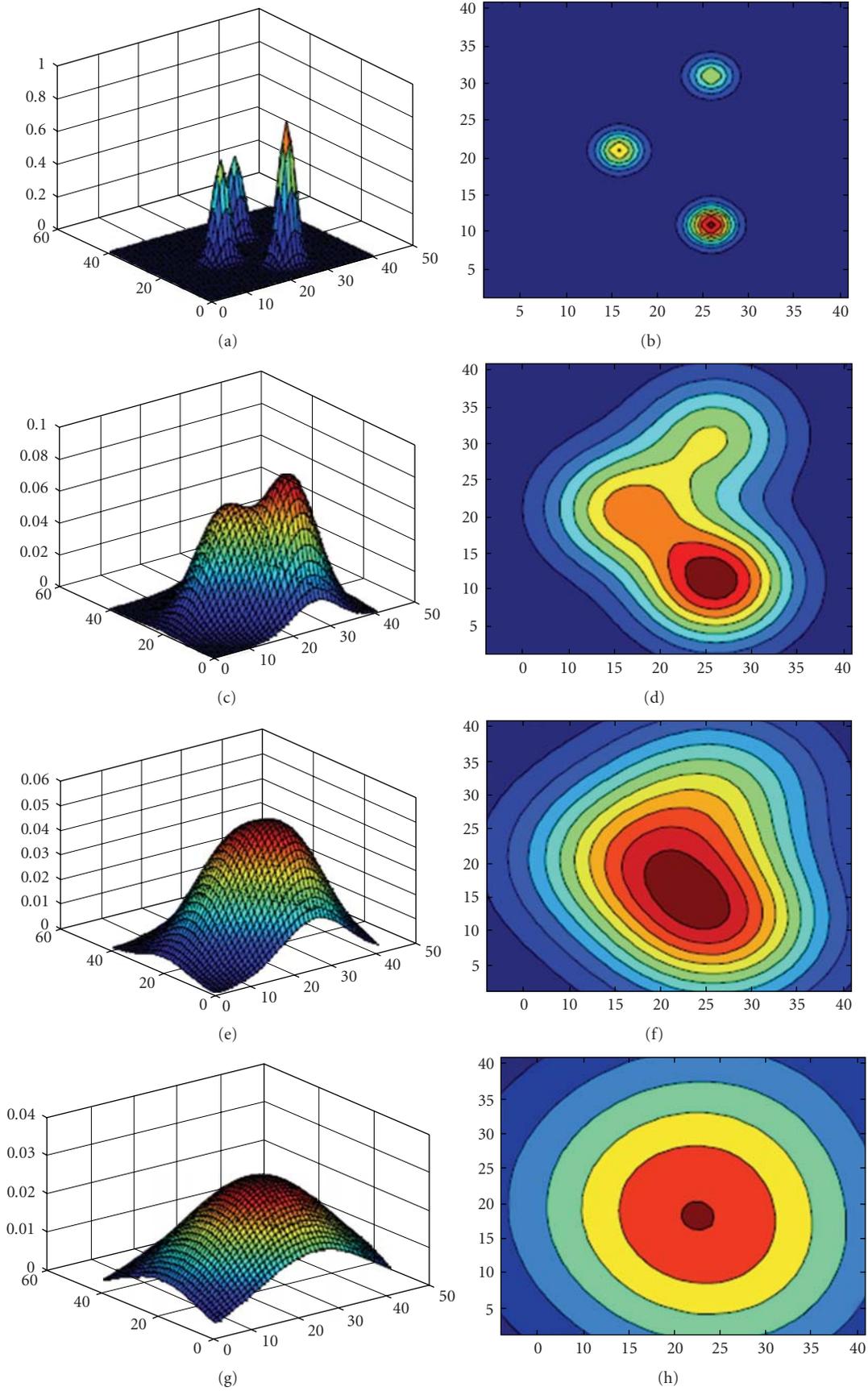


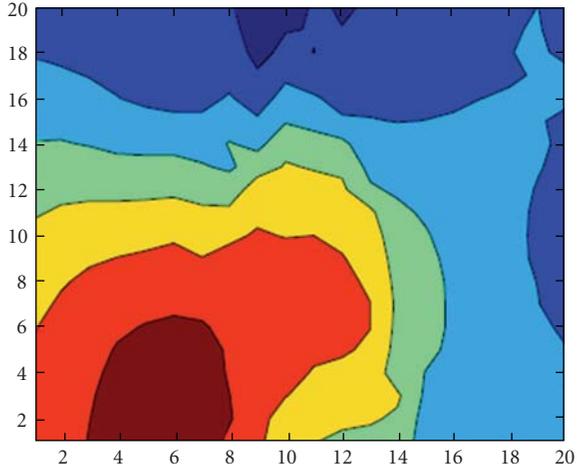
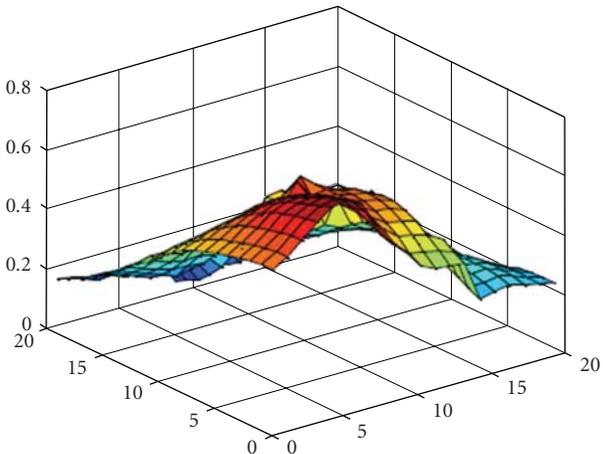
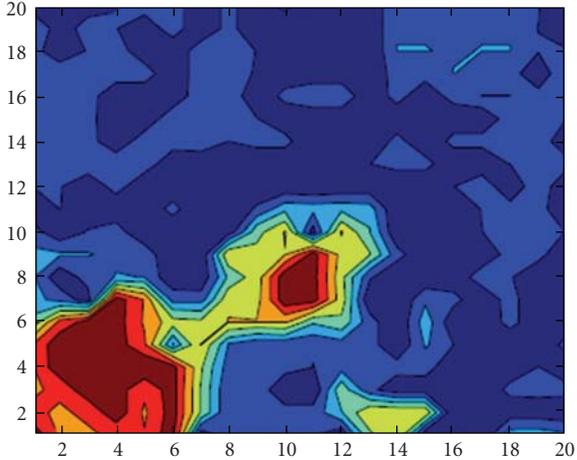
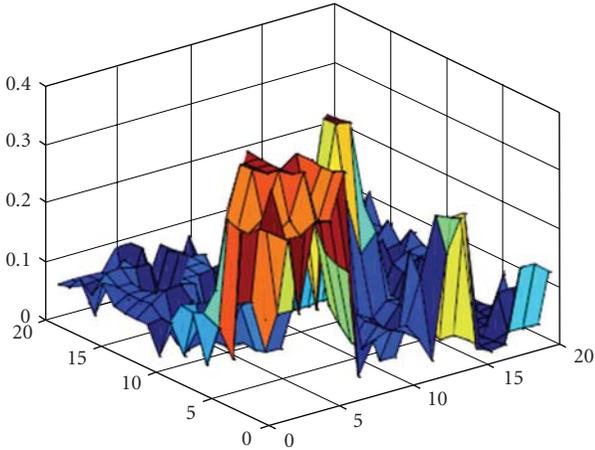
FIGURE 2: Two-dimensional Gaussian mixture surface changes with the bandwidth: $[0.13 \ 0; 0 \ 0.13]$, $[1.3 \ 0; 0 \ 1.3]$, $[2.6 \ 0; 0 \ 2.6]$, and $[6.5 \ 0; 0 \ 6.5]$.



Frame 45

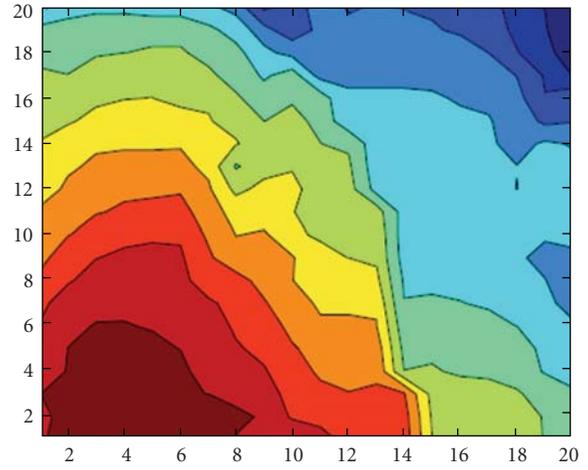
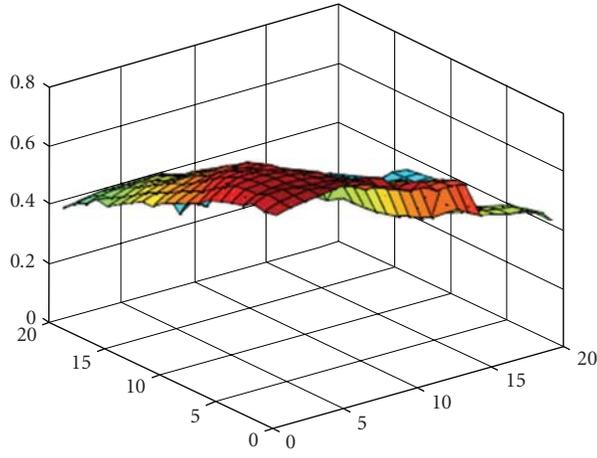


Frame 55

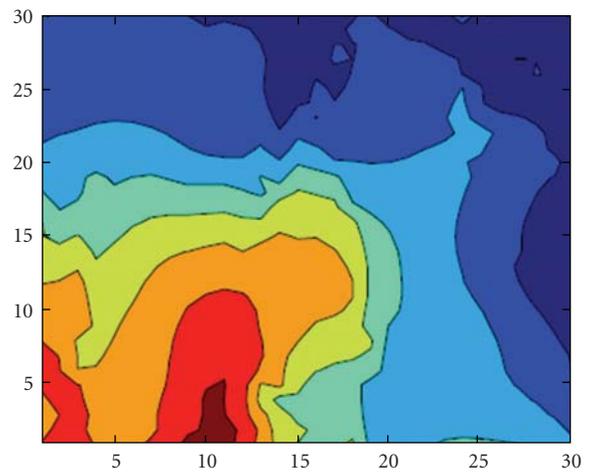
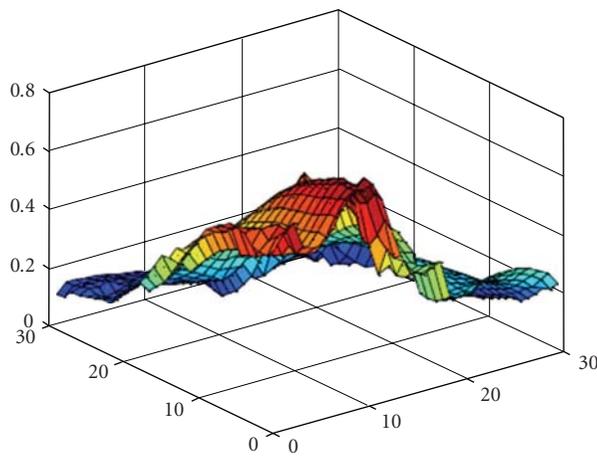
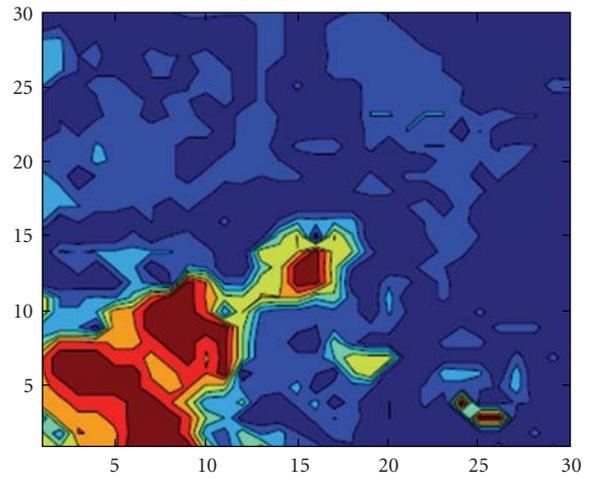
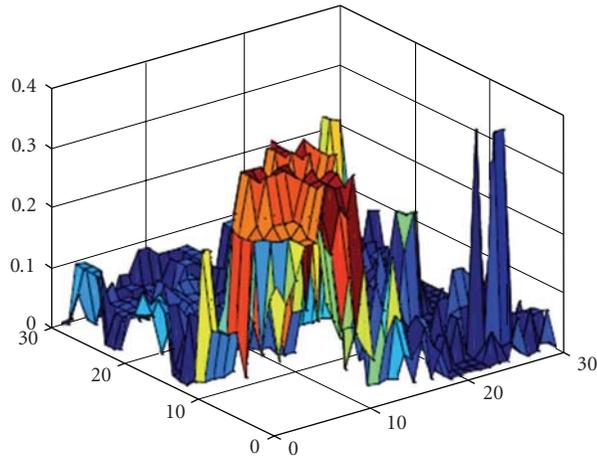


(a)

FIGURE 3: Continued.

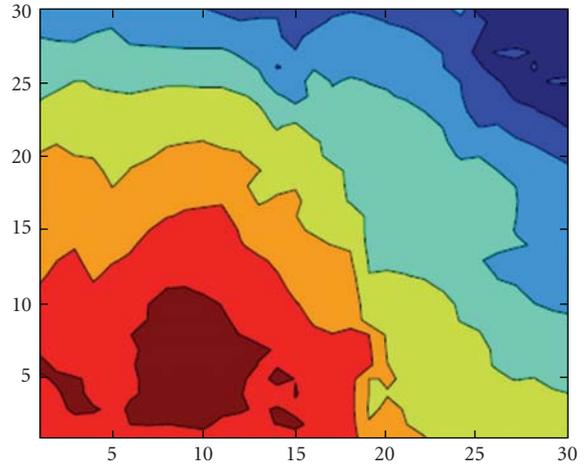
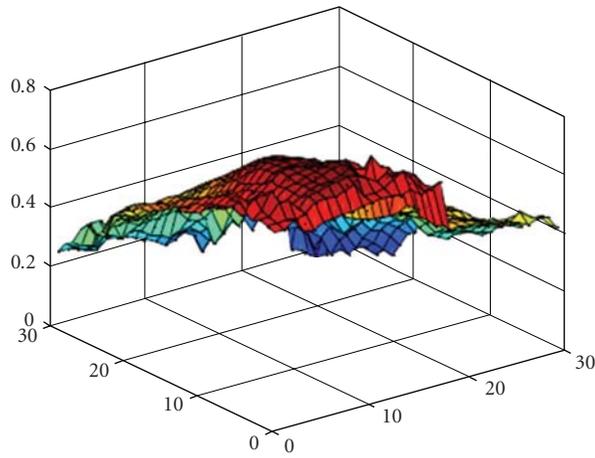


(a) The bandwidth increase effect on the location of modes in an optimization surface with resolution 10*10 between frames 45 and 55

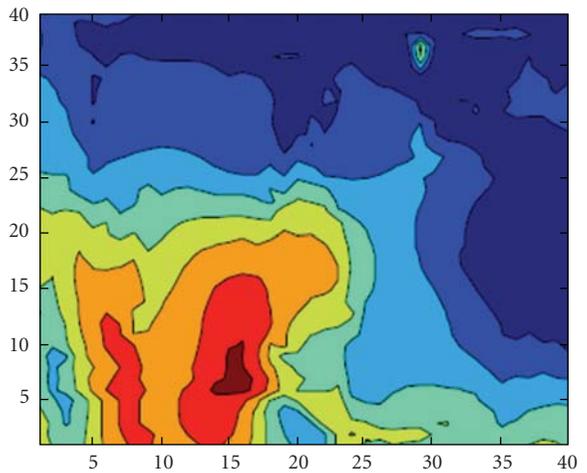
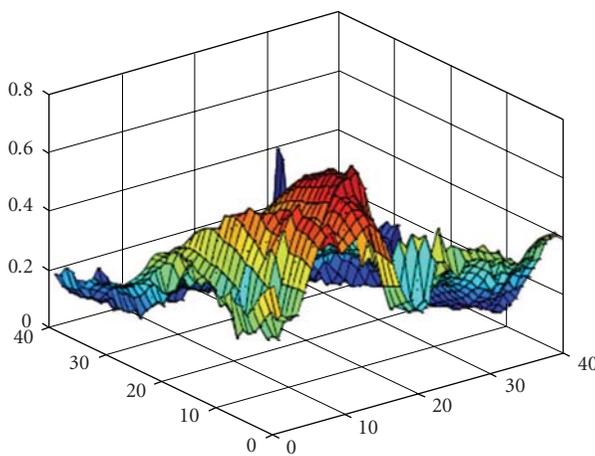
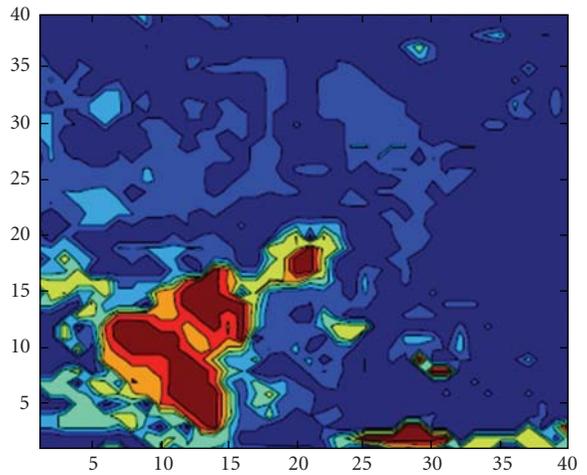
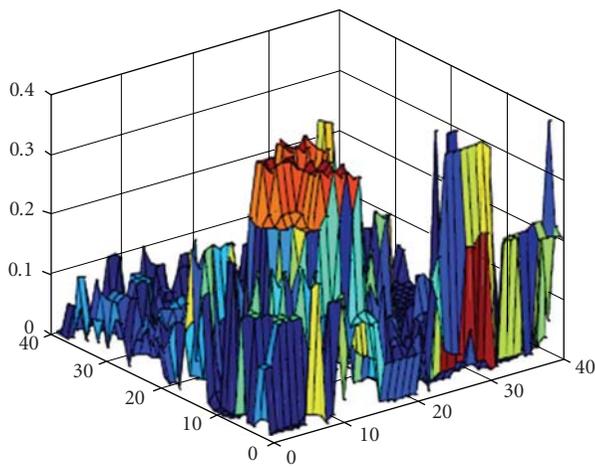


(b)

FIGURE 3: Continued.

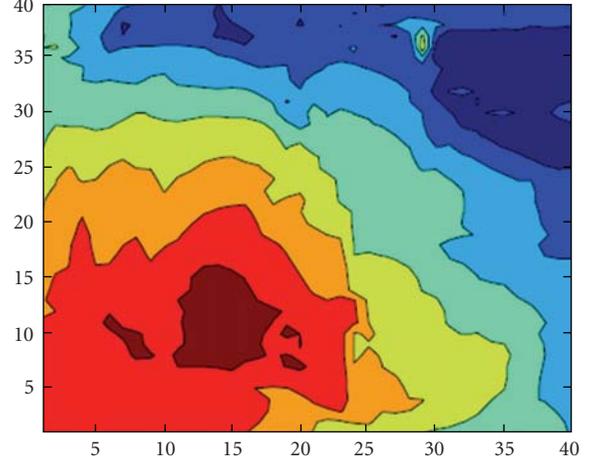
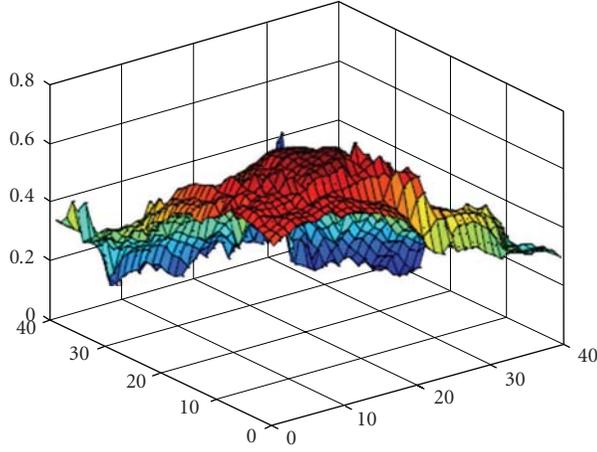


(b) The bandwidth increase effect on the location of modes in an optimization surface with resolution 20*20 between frames 45 and 55



(c)

FIGURE 3: Continued.



(c) The bandwidth increase effect on the location of modes in an optimization surface with resolution 30*30 between frames 45 and 55

FIGURE 3

robust. Because of the possibility of overshooting due to sufficiently large step-size occurrence this method can also be utilized and adopted in clustering through segmentation, since it is potential enough to gather patterns into a cluster as a global and local mode.

5. Tracking Application and Implementation

Implementing the algorithm and the experimental result on an object is represented in a normalized rectangle region as the target region. We choose color as feature in target model p and target candidate q region. As it is clear, we should measure the similarity between the two selected regions in two consecutive frames through similarity measurement criteria $\hat{\rho}(x) = \rho(\hat{q}, \hat{p}(x))$ such as Bhattacharyya, Kullback-Leibler or Matusita, respectively in [2, 4, 5, 40, 41]. The pixel of rectangle region $\{y_i\}_{i=1}^n$ are being assigned weight according to their distance from the center of normalized circle surrounded in the rectangle, less weight and value they have through Gaussian function distribution. Via the weighted pixels, their colors in RGB scale will be extracted. Then the color would be sorted into m Bin histogram with their weight, consequently their value would be summed up in their R, G and B color bins of the histogram. This process is followed in both target model and candidate regions

$$\hat{p}_u(x) = \frac{1}{c} \sum_{i=1}^n k(\|y_i\|^2) \delta((b(y_i) - u)), \quad (6)$$

p and q are the normalized density value of the target model and candidate. Each pixel value $\{y_i\}_{i=1}^n$ is converted into histogram bin via its color accompanied by its weight. The similar process is performed upon the both target model and candidate. With the initial point x the tracking procedure started through approximating the distance between the

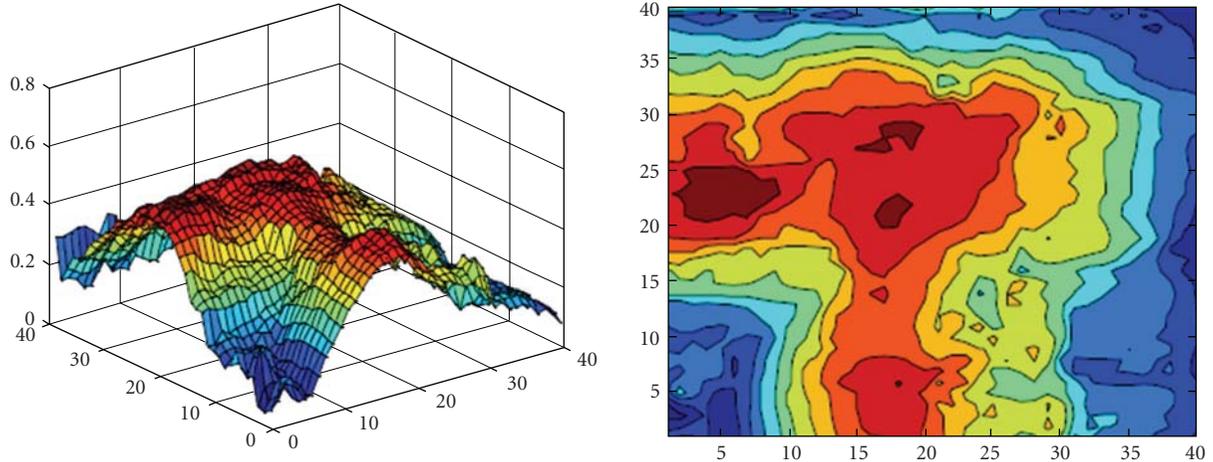
place and location of the object in the two consecutive frames Δx . Then the new location of object would be found and object is tracked

$$\Delta x^* = \arg \max_{\Delta x} \rho(\hat{q}, \hat{p}(x^* + \Delta x)). \quad (7)$$

Using the Taylor expansion, the linear first-order extension helps us solve the optimization problem efficiently via MS iteration at the initial point x through extending similarity [2, 5].

5.1. Location Approximation. Conventional MS cannot seek a global mode in presence of local mode due to the fixed-bandwidth which is created by rapid motion, illumination changes, clutter and occlusion as shown in Figure 3. In our experiment, it does not matter for improved MS the initial point to start. It can easily locate the target fast. In the examples, there may be different number of bandwidth but entirely they are 4 or 3 for all tracking sequences as clarified in Figure 3. We are using color as feature but of course it is possible to be taking advantage of other features as well like motion and orientation. If we do not use the multi-bandwidth procedure, most of the searches will end to local modes unless they are initially located at the closest basin points to the global mode in Figure 3. Using the broad bandwidth kernel function at first MS run will let us find this closest basin point near to the global mode.

5.2. Object Location and Track. Entirely all of the methods in object tracking, ever written and proposed could have covered some of the weak points in this field, but there are some common problems in all of them: All have problem in finding the object location in large distance movement between two successive frames. The starting-point for them



(a) Top is cost function surface between frame 91 and 92. Black-colored clutter in background, below images, created some local modes in the surface



(b) Frames 88, 89, 90, 91, and 92 show us a tracking failure due to the black-colored clutter in background

FIGURE 4: Top is cost function surface between frame 91 and 92.

all is very important to track the object correctly. Background problem such as clutter, occlusion, and illumination changes can completely influence the tracking path and cause failure exactly as illustrated in Figure 4. They are not really capable to self-recovering from their failure due to the same local modes existing in Figure 4(a). It failed in tracking as shown in Figure 4(b).

The conventional MS has all of these problems described above as well. In our work, we have enabled the tracker to be robust enough in different initial points in an image by considering the efficiency and efficacy of bandwidth variation of kernel function through adaptive step-size iteration.

We are actually utilizing an object detector incorporated in localizing procedure to recover from any failure when occurred. It was also previously proposed to use a detector for Particle filtering tracking-based [42]. By choosing a broad bandwidth we try to pass other local modes to reach the basin of attraction for the desired global mode. Some implementation problems:

The multi-bandwidth tracker starts in a 3 or 4 bandwidth shifting iteration in an MS procedure. It is worth to say that through using color as feature so that there may be some unwanted modes created just because of the difference between two points color values [43] which may cause a local

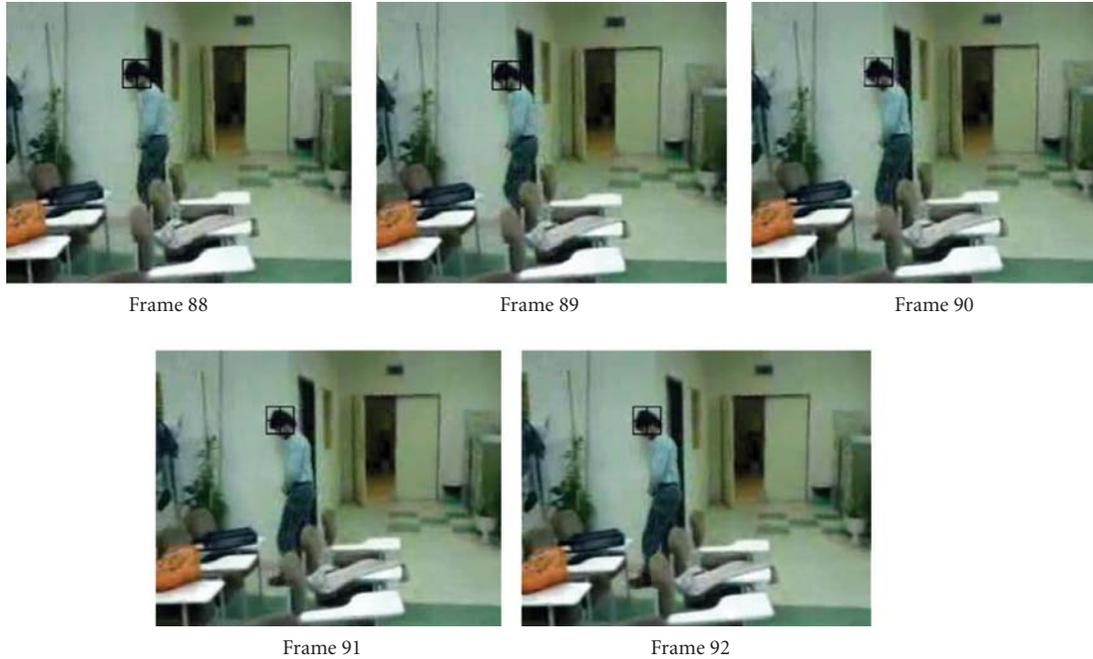


FIGURE 5: Multi-bandwidth tracker did not fail to track in this cluttered sequence (multimodal surface).

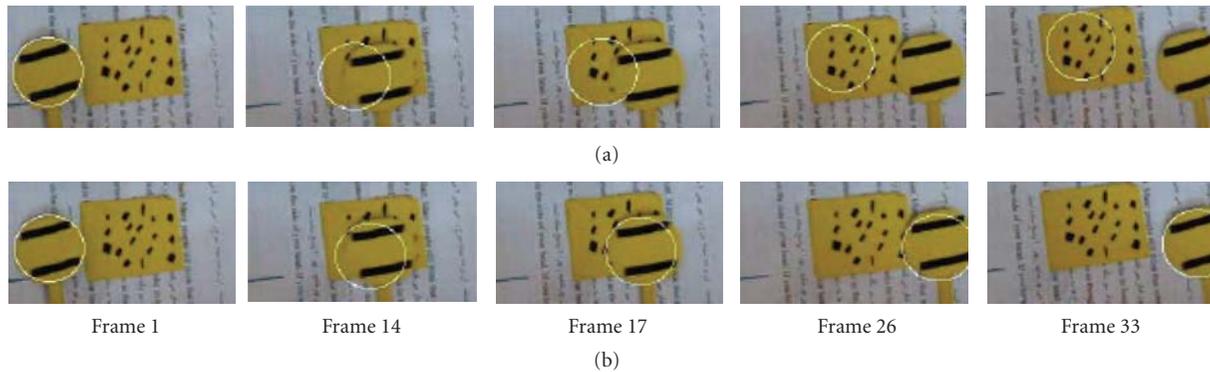


FIGURE 6: Top row: MS, Bottom Row: MBMS.

mode that cause the MS seeker to be trapped in its basin of attraction. Through a modification in color histogram value, we can increase the step-size adaptively. If we want to have faster mode seeking so in a trade off balance we may lose some accuracy as well. Recently it has been proposed that we can also add some accuracy with a little more time of computation [44]. We have to increase our computational cost due to larger bandwidth kernel windowing in a frame. As explained, this algorithm was applied to application of an object tracking with faster mode seeking results [36] as shown in Table 1(a) and 1(b).

The most important problem of the proposed method is that the series of bandwidth selection is manual, but we can be looking for some issues to be proposing an automatic selector of bandwidth using some features, but in this paper, we are using manually multi-bandwidth series to track correctly as illustrated in Figure 5.

Figure 6. In this circle tracking scenario, MS fails to track the circle due to getting distracted by the dotted square (i.e., a local mode) but in the same Frame(17) while the MBMS (i.e., the proposed method) is successfully tracking the circle in spite of square distraction in that frame. As can be seen in (Frame 26), MS tracker is unable to track successfully as the circle is moved in front of the square.

Figure 8. In this Bus tracking scenario, MS fails to track the Bus due to getting distracted by the similar color clutters (i.e., two local modes) while the MBMS (i.e., the proposed method) is successfully tracking the bus in spite of clutter distraction in these frames. As can be seen in MS tracker is unable to track successfully as the bus is moved in front of the same color clutters.

Figure 10. In this hand tracking scenario, MS (a) fails to track the hand due to getting distracted by the face (i.e., a local mode) but in the same frame (140) while the MBMS (i.e., the proposed method) (b) is successfully tracking the

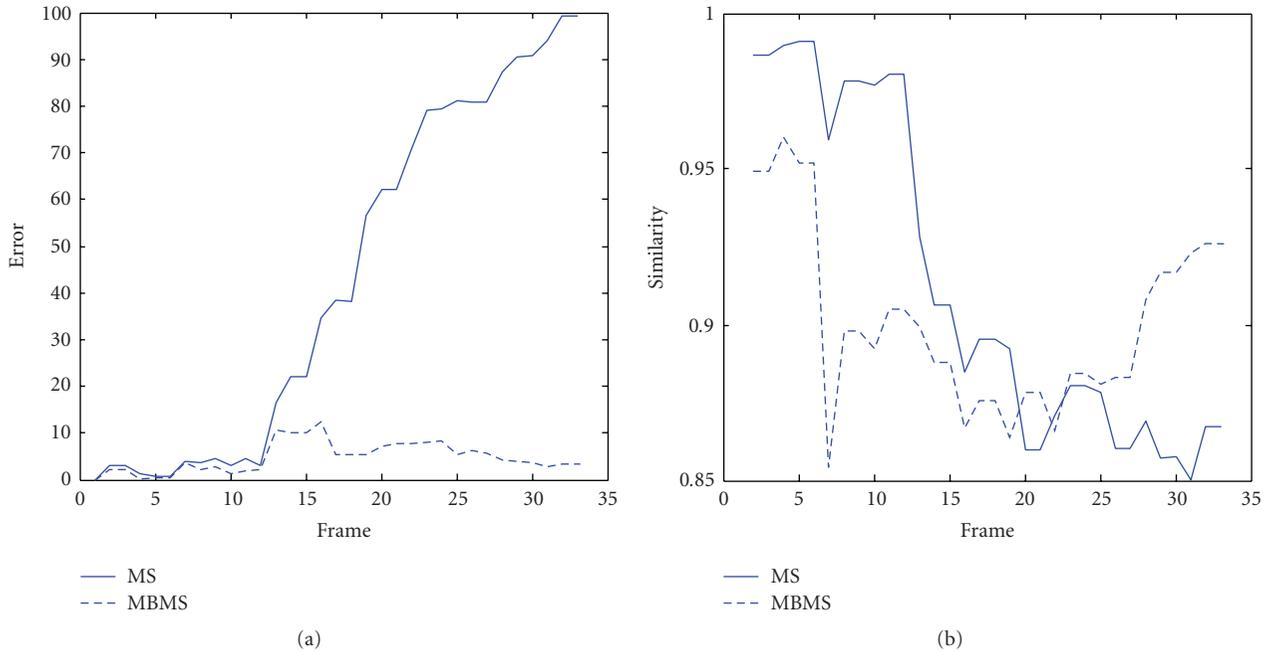


FIGURE 7: Left plot: Error from the GT (i.e., Ground Truth is zero state), Right plot: Similarity with Target (GT has 100% similarity). Compared to the real GT (i.e., Ground Truth), we can observe that MBMS successfully performs circle tracking through the entire sequence.

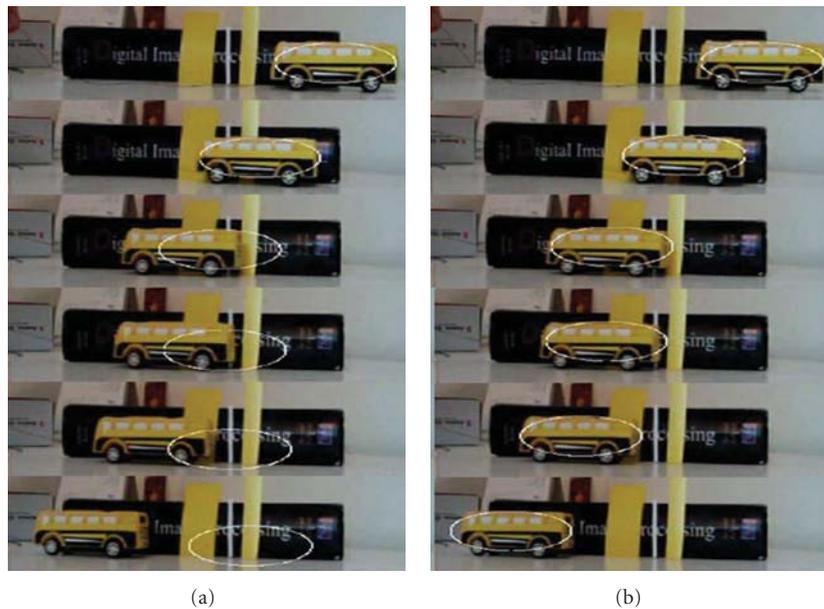


FIGURE 8: Frames: 1, 52,79,82,90,110, left column: MS, right column: MBMS.

hand in spite of face distraction in that frame. As can be seen in ((a)-frame 260) MS tracker is unable to track successfully as the hand is moved in front of the face. Compared to the real GT (i.e., Ground Truth) in (c), we can observe that MBMS successfully tracks the hand through the entire sequence.

Definitely, we can observe that MBMS successfully performs better hand tracking through the entire sequence with lower number of iterations than MS.

At Table 2-Comparison of Number of Iterations for Convergence for 1D and 2D data set. The Initial Location For Each Run Is Shown in the Second Column.

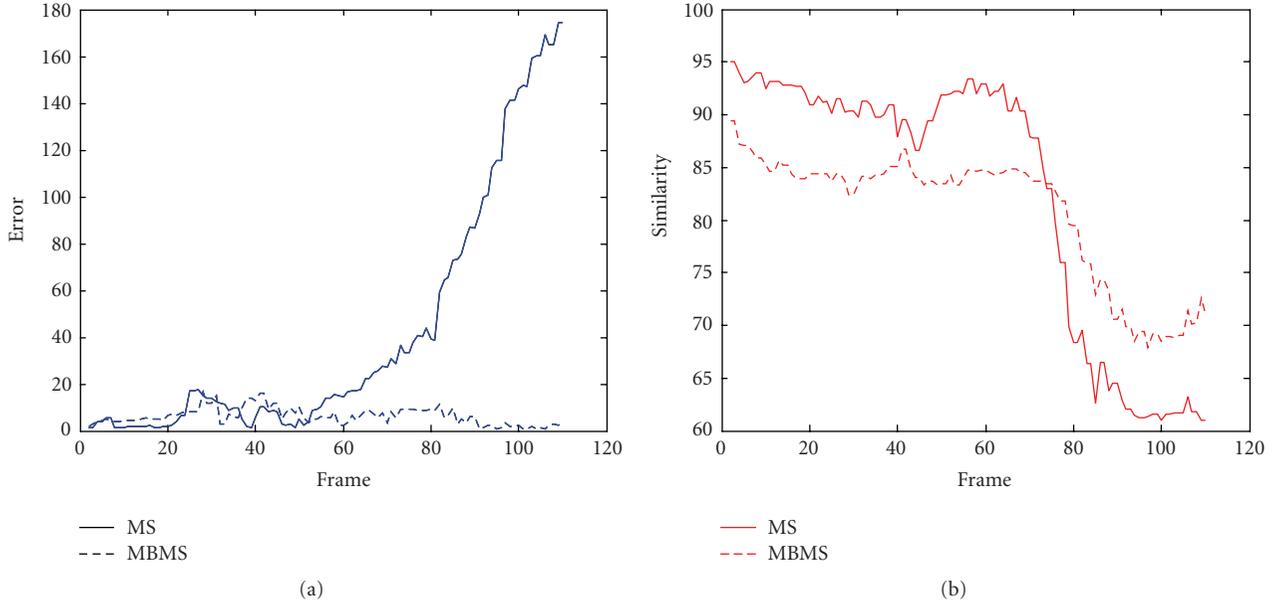


FIGURE 9: Left plot: Error from the GT (i.e., Ground Truth is zero state), Right plot: Similarity with Target (GT has 100% similarity). Compared to the real GT (i.e., Ground Truth), we can observe that MBMS successfully performs bus tracking through the entire sequence.

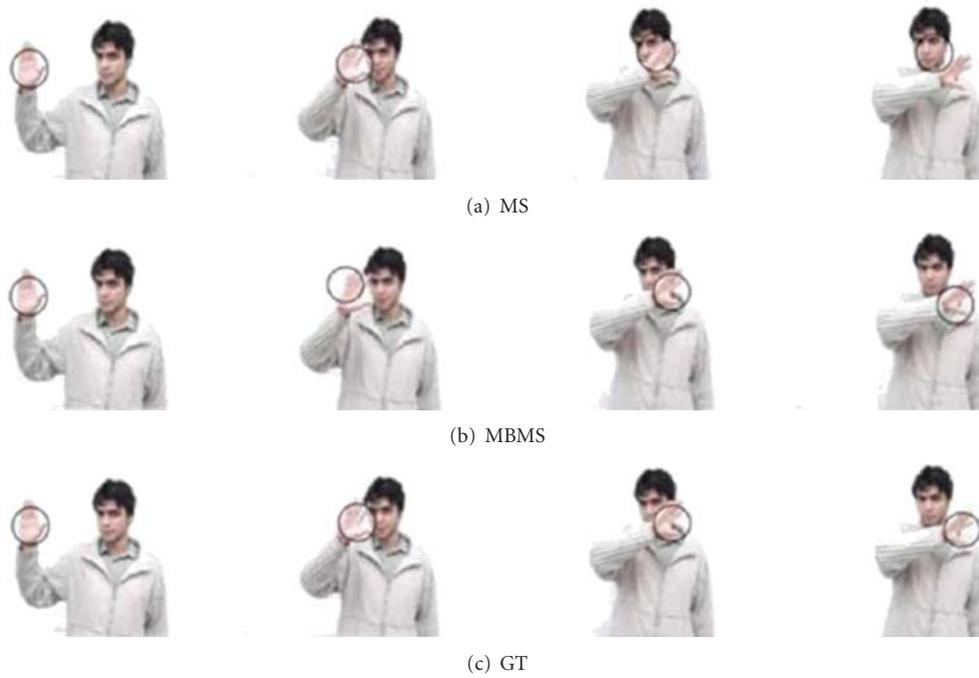


FIGURE 10: Frames: 1, 140, 260, 290.

(1) Data set no. 1 (1D synthetic data). A total of 1000 data points are drawn with equal probability from four normals: $N(3,1)$, $N(1,1)$, $N(0,1)$, and $N(-2,1)$

(2) Data set no. 2 (2D synthetic data). A total of 1050 bivariate data points are drawn with equal probability from three normals

$$N\left(\begin{bmatrix} -7 \\ 10 \end{bmatrix}, \begin{bmatrix} 5.5 & -4.5 \\ -4.5 & 5.5 \end{bmatrix}\right), \quad N\left(\begin{bmatrix} 0 \\ 12 \end{bmatrix}, \begin{bmatrix} 8.5 & 6.5 \\ 6.5 & 8.5 \end{bmatrix}\right), \\ N\left(\begin{bmatrix} 12 \\ 15 \end{bmatrix}, \begin{bmatrix} 14.4 & -4.5 \\ -4.5 & 3.6 \end{bmatrix}\right). \quad (8)$$

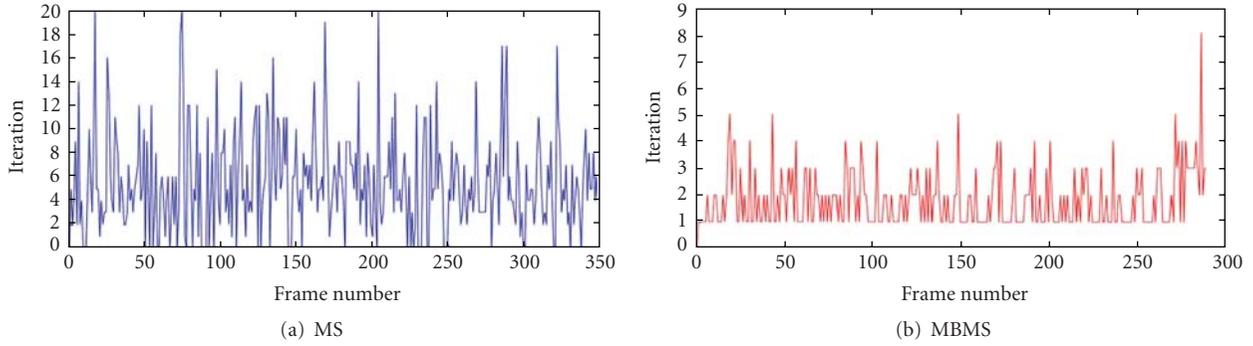


FIGURE 11: Top plot: Iteration number for MS while tracking the hand in Figure 10(a), Bottom plot: Iteration number for MBMS while tracking hand in Figure 10(b).

TABLE 1

(a) The step size iterations between frame 88 and 89.

Algorithm	Step-size x	Step-size y	Center in Frame 88	Center in Frame 89
Standard MS	0.0040	-0.0006	106	106
Multi-Bandwidth	0.0027	-0.007	58	58

(b) It shows the step size iterations between frame 89 and 90 in seven iteration but different step sizes for standard method and proposed one.

Algorithm	Step-size		Center position of square 90	
	x	y	x	y
Standard	-0.1353	0.0006	104.5116	58.0067
Multi-bandwidth	-0.0889	-0.0192	105.0216	57.7888
Standard	-0.0881	-0.0002	103.5420	58.0048
Multi-bandwidth	-0.0640	-0.0199	104.3180	57.5700
Standard	-0.0515	0.0200	102.9758	58.2248
Multi-bandwidth	-0.0449	-0.0037	103.8240	57.5298
Standard	-0.0754	0.0185	102.1470	58.4286
Multi-bandwidth	-0.0613	-0.0138	103.1495	57.3778
Standard	-0.0082	0.0169	102.0563	58.6143
Multi-bandwidth	-0.0278	-0.0121	102.8439	57.2442
Standard	-0.0197	0.0023	101.8392	58.6395
Multi-bandwidth	-0.0265	-0.0193	102.5524	57.0324
Standard	0.0124	0.0181	101.8392	58.6395
Multi-bandwidth	-0.0101	-0.0107	102.5524	57.0324

TABLE 2: Comparison of Number of Iterations for Convergence for 1D and 2D data set.

Data set	Initial	Number of iterations	
		MBMS	MS
data set #1	0	33	51
	1	36	77
	3	21	33
data set #2	(-5,20)	22	34
	(-10,16)	21	29
	(20,10)	23	35

6. Conclusion

A new kernel-based object tracking framework is proposed. The contribution is mainly the use of a prior large bandwidth for a priori tracking followed by the estimated tracking. This

framework is robust to noise and clutters so that it can escape from many local maxima. This tracking algorithm (i.e., MBMS) can converge faster than does the conventional kernel-based object tracking (i.e., MS). However, there are still some problems, and some weaknesses which are to

be later clarified and rephrased. Many results can be analyzed theoretically. This paper as a reference can be much helpful for later extension of this work. The experimental results above must have illustrated this approach performance. As shown in above database, it can also be concluded that in rapid motion of an object, large displacement between two adjacent frames occurs which will lead MS to a failure in tracking an object. By means of multi-bandwidth proposal, we can be improving MS in recovering from the failure by incorporating a detector in localization process called multi-bandwidth kernel functionality. In comparison with conventional MS and other techniques like [1] and fast transforming the Gaussian mixture [7, 29], the speed of convergence has increased and the number of iterations has decreased in contrast with an enhancement in each step-size iteration. Object tracking is an important issue in Artificial Intelligence. Its worldwide usages in robotic engineering, Machine Intelligence, Computer Vision, and Human-Computer Interface (HCI) are well-known throughout the world of engineering sciences. In the future, this method can be extended to the more automatic bandwidth selector equipped with several other features to track objects in many varieties of applications inside the industry.

References

- [1] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: a texture classification example," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 456–463, Nice, France, October 2003.
- [2] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.
- [3] R. T. Collins, "Mean-shift blob tracking through scale space," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, pp. 234–240, Madison, Wis, USA, June 2003.
- [4] A. Elgammal, R. Duraiswami, and L. S. Davis, "Probabilistic tracking in joint feature-spatial spaces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, pp. 781–788, Madison, Wis, USA, June 2003.
- [5] G. D. Hager, M. Dewan, and C. V. Stewart, "Multiple kernel tracking with SSD," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, pp. 790–797, Washington, DC, USA, July 2004.
- [6] H. Wang and D. Suter, "MDPE: a very robust estimator for model fitting and range image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 139–166, 2004.
- [7] C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 176–183, San Diego, Calif, USA, June 2005.
- [8] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with application in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.
- [9] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.
- [10] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [11] M. Fashing and C. Tomasi, "Mean shift is a bound optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 471–474, 2005.
- [12] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.
- [13] É. Arnaud, É. Mémin, and B. Cernuschi-Frías, "Conditional filters for image sequence-based tracking—application to point tracking," *IEEE Transactions on Image Processing*, vol. 14, no. 1, pp. 63–79, 2005.
- [14] Z. Zivkovic and B. Kröse, "An EM-like algorithm for color-histogram-based object tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, pp. 798–803, Washington, DC, USA, July 2004.
- [15] Z. Fan, Y. Wu, and M. Yang, "Multiple collaborative kernel tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 502–509, San Diego, Calif, USA, June 2005.
- [16] P. van Laarhoven and E. Aarts, *Simulated Annealing: Theory and Applications*, Springer, New York, NY, USA, 1987.
- [17] R. M. Neal, "Annealed importance sampling," *Statistics and Computing*, vol. 11, no. 2, pp. 125–139, 2001.
- [18] J. Deutscher and I. Reid, "Articulated body motion capture by stochastic search," *International Journal of Computer Vision*, vol. 61, no. 2, pp. 185–205, 2005.
- [19] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [20] K. Zhang, M. Tang, and J. T. Kwok, "Applying neighborhood consistency for fast clustering and kernel density estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 1001–1007, San Diego, Calif, USA, June 2005.
- [21] C. Yang, R. Duraiswami, D. DeMenthon, and L. Davis, "Mean-shift analysis using quasi-newton methods," in *Proceedings of the International Conference on Image Processing (ICIP '03)*, pp. 447–450, Barcelona, Spain, September 2003.
- [22] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Computation*, vol. 8, no. 1, pp. 129–151, 1996.
- [23] L. E. Ortiz and L. P. Kaelbling, "Accelerating EM: an empirical study," in *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 512–521, Stockholm, Sweden, 1999.
- [24] R. Salakhutdinov and S. Roweis, "Adaptive overrelaxed bound optimization methods," in *Proceedings of the 20th International Conference on Machine Learning*, pp. 664–671, Washington, DC, USA, August 2003.
- [25] C. Shen, M. J. Brooks, and A. van den Hengel, "Fast global kernel density mode seeking with application to localisation and tracking," in *Proceedings IEEE International Conference on Computer Vision*, vol. 2, pp. 1516–1523, Beijing, China, October 2005.
- [26] D. Comaniciu, "An algorithm for data-driven bandwidth selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 281–288, 2003.
- [27] M. C. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," *Journal of the*

- American Statistical Association*, vol. 91, no. 433, pp. 401–407, 1996.
- [28] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz, *Non-parametric and Semiparametric Models*, Springer Series in Statistics, Springer, New York, NY, USA, 2004.
- [29] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, “Improved fast gauss transform and efficient kernel density estimation,” in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 464–471, Nice, France, October 2003.
- [30] P. Hall, M. C. Minnotte, and C. Zhang, “Bump hunting with non-Gaussian kernels,” *Annals of Statistics*, vol. 32, no. 5, pp. 2124–2141, 2004.
- [31] P. Chaudhuri and J. S. Marron, “Scale space view of curve estimation,” *Annals of Statistics*, vol. 28, no. 2, pp. 408–428, 2000.
- [32] T. Lindeberg, *Scale-Space Theory in Computer Vision*, Kluwer, Norwell, Mass, USA, 1994.
- [33] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, Mass, USA, 1987.
- [34] D. Comaniciu, “Nonparametric information fusion for motion estimation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’03)*, vol. 1, pp. 59–66, Madison, Wis, USA, June 2003.
- [35] M. Carreira-Perpiñán and C. Williams, “On the number of modes of a Gaussian mixture,” Tech. Rep. EDI-INF-RR-0159, School Informatics, University of Edinburgh, Edinburgh, UK, February 2003, <http://www.inf.ed.ac.uk/publications/report/0159.html>.
- [36] C. Shen and M. J. Brooks, “Adaptive over-relaxed mean shift,” in *Proceedings of the 8th International Symposium on Signal Processing and Its Applications*, Sydney, Australia, August 2005.
- [37] S. Hettich and S. D. Bay, “The UCI KDD archive,” Tech. Rep., Department of Information and Computer Science, University of California, Irvine, Calif, USA, 1999, <http://kdd.ics.uci.edu/>.
- [38] T.-L. Liu and H.-T. Chen, “Real-time tracking using trust-region methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 397–402, 2004.
- [39] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, 1989.
- [40] T. Kailath, “The divergence and Bhattacharyya distance measures in signal selection,” *IEEE Transactions on Communication Technology*, vol. 15, no. 1, pp. 52–60, 1967.
- [41] S. Kullback and R. A. Leibler, “On information and sufficiency,” *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [42] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, “A boosted particle filter: multitarget detection and tracking,” in *Proceedings of the 8th European Conference on Computer Vision*, vol. 1, pp. 28–39, Prague, Czech Republic, May 2004.
- [43] H. Wang and D. Suter, “False-peaks-avoiding mean shift method for unsupervised peak-valley sliding image segmentation,” in *Proceedings of the 7th International Conference on Digital Image Computing: Techniques and Applications (DICTA ’03)*, pp. 581–590, Sydney, Australia, December 2003.
- [44] A. T. Kalai and S. Vempala, “Simulated annealing for convex optimization,” *Mathematics of Operations Research*, vol. 31, no. 2, pp. 253–266, 2006.

Research Article

3D Medical Volume Segmentation Using Hybrid Multiresolution Statistical Approaches

Shadi AlZu'bi¹ and Abbas Amira²

¹ Department of Electronic & Computer Engineering, School of Engineering and Design, Brunel University, West London, Uxbridge UB8 3PH, UK

² Nanotechnology and Integrated BioEngineering Centre (NIBEC), Faculty of Computing and Engineering, University of Ulster, Shore Road Newtownabbey Co. Antrim BT37 0QB, Northern Ireland

Correspondence should be addressed to Shadi AlZu'bi, shadi.alzubi@brunel.ac.uk

Received 8 February 2010; Revised 24 May 2010; Accepted 24 June 2010

Academic Editor: Chu-Song Chen

Copyright © 2010 S. AlZu'bi and A. Amira. 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.

3D volume segmentation is the process of partitioning voxels into 3D regions (subvolumes) that represent meaningful physical entities which are more meaningful and easier to analyze and usable in future applications. Multiresolution Analysis (MRA) enables the preservation of an image according to certain levels of resolution or blurring. Because of multiresolution quality, wavelets have been deployed in image compression, denoising, and classification. This paper focuses on the implementation of efficient medical volume segmentation techniques. Multiresolution analysis including 3D wavelet and ridgelet has been used for feature extraction which can be modeled using Hidden Markov Models (HMMs) to segment the volume slices. A comparison study has been carried out to evaluate 2D and 3D techniques which reveals that 3D methodologies can accurately detect the Region Of Interest (ROI). Automatic segmentation has been achieved using HMMs where the ROI is detected accurately but suffers a long computation time for its calculations.

1. Introduction

Volume segmentation allocates the voxels in 3D images into partitions or 3D regions that represent meaningful physical entities. The goal is to distinguish between different regions in the 3D volume and cover the extracted contours from the entire volume. Voxels' classification into regions is performed according to a certain region to which the voxels belong, and some shared, predefined properties. Those voxels comprise an isolated or segmented Object Of Interest (OOI) from the input volume.

There are many existing techniques used for medical image segmentation, including Multiresolution Analysis (MRA), statistical methods, and thresholding- and clustering-based techniques. Clustering technique aims to classify each pixel in an image into the proper cluster, and then these clusters are mapped to display the segmented images. A certain clustering criterion can be adopted to group each pixel into a specific number of clusters depending

on the image histogram [1, 2]. Medical images can also be segmented using thresholding approaches by partitioning their intensities. When images contain different structures with contrasting intensities, thresholding provides a simple but effective means for obtaining segmentation. Generally, the thresholds are generated based on visual assessment of the resulting segmentation [3, 4].

MRA allows the preservation of an image according to certain levels of resolution. Consequently, wavelets have been useful in image compression, de-noising, and classification. Wavelet theory, which is built on solid mathematical foundations uses well-established tools such as quadrature mirror filtering, subband coding, and pyramidal image processing. Wavelet analysis enables the exploitation of signal or image characteristics associated with a particular resolution level, which may not be detected using other analysis techniques [5, 6].

Statistical modeling is a set of mathematical equations which describes the behavior of an object of study in

terms of random variables and the associated probability distribution. Markov Random Field Model (MRFM) is a statistical approach which has been utilized within segmentation methodologies to model spatial interactions between neighbor pixels [1, 3, 7]. These local correlations between pixels are used as a mechanism for modeling various image properties. From a medical imaging perspective, this kind of analysis is useful, as most pixels can be categorized into the same class as their neighbors [1].

Statistical models using Hidden Markov Models (HMMs) observe a sequence of emissions with a hidden sequence of states that the model went through to generate the emissions [8, 9]. HMM states are not directly visible to the observer, but the variables influenced by the states are visible. Each state in HMM has a probability distribution over the other states which evaluate the state sequence. The challenge in HMM is to determine the hidden parameters from the observable parameters to be used in performing further analysis [8, 9].

This paper focuses on the implementation of efficient and robust medical volume segmentation techniques. MRA including wavelet and ridgelet transforms have been deployed for feature extraction, while statistical modeling using HMMs has been used for segmentation. The outline of this paper is as follows. In the following section, the proposed segmentation system is illustrated and discussed. In Section 3, the mathematical background of the proposed segmentation methods is presented with some test images. Section 4 presents the results and analysis of the proposed techniques for medical volumes segmentation. Conclusions and implications for future work are discussed in Section 5.

2. Proposed Segmentation System

In medical applications, the source of the 3D data set is the acquisition systems such as Positron Emission Tomography (PET), Computerized Tomography (CT), or Magnetic Resonance Imaging (MRI). Such devices are capable of slicing an object in a physical sectioning. 3D data set from those devices can be considered as parallel slices stacked to form a 3D volume. A segmented medical volume into sub-volumes which are more meaningful and easier to analyze and understand is the output the proposed system illustrated in Figure 1.

Hybrid multiresolution statistical approaches and other segmentation techniques are used to achieve accurate segmented volumes. System input is a 3D phantom or real volume from scanner acquisition. Acquisition systems produce a number of 2D slices resulted from the scanned volume of the body. These slices can be individually segmented using 2D segmentation methods such as thresholding, clustering, and HMMs followed by volume reconstruction or directly using 3D segmentation methods such as 3D that thresholding or 3D discrete wavelet transform (3D-DWT) after volume reconstruction process.

This paper explains the new application of wavelet transform directly on the 3D medical volumes from the acquisition systems using 3D-DWT with Haar wavelet filter.

HMMs have been also applied on those volumes slice-by-slice to segment the Region Of Interest (ROI) into a number of classes based on the grey scale values of the original volume pixels.

3. Segmentation Methods and Their Mathematical Backgrounds

The mathematical background of the developed techniques for 3D medical volume segmentation system is presented in this section.

3.1. Thresholding. Scalar images can be segmented using thresholding approaches by partitioning image intensities. This methodology attempts to determine an intensity value that can separate the signal into a desired number of classes. Segmented images can be achieved by clustering all pixels with intensities larger than the threshold value into one class, and all others into another. In many applications, the threshold values selection can be done depending on the basis of histogram. Multithresholding occurs when more than one threshold value is determined [10].

The voxels of a certain object are not necessarily connected after thresholding because this technique does not consider the spatial characteristics of an image, thus causing it to be sensitive to noise and intensity fluctuations. For this reason it cannot be easily applied to many medical imaging modalities. These drawbacks essentially corrupt the histogram of the image-making partitioning via the selection of more problematic appropriate thresholds [11].

Hard thresholding technique is a boolean filter [10] which depends on pixel value and threshold value. As illustrated in Algorithm 1, it either makes the input zero or keeps it without any changes [12].

Hard thresholding process is less complex than soft thresholding (Algorithm 2), where pixels that have values greater than threshold value do not change [13]; soft thresholding replaces each pixel which has greater value than the threshold value by the difference between threshold value and pixel value [10]. This makes the process more complicated and increases the processing time for the algorithm. Figure 2 illustrates a segmented medical brain slice from an MRI scanner using hard and soft thresholding techniques. Many other thresholding types are widely used in different areas including adaptive thresholding.

It can be seen here that applying thresholding techniques is a very easy process and can be affected easily by surrounding noise, but it has been used as a preprocessing step and postprocessing step with other segmentation techniques. It is worth mentioning that thresholding can be replaced by the clustering technique which will be explained in Section 3.2.

3D thresholding method is similar to the 2D approaches where thresholding process is applied on all pixels in the volume instead of that in the plane. Algorithm 3 explains the pseudocode for 3D thresholding.

3.2. Clustering. Clustering technique is the process of classifying each group of pixels in an image into one class, each class has the same or similar properties which evaluate

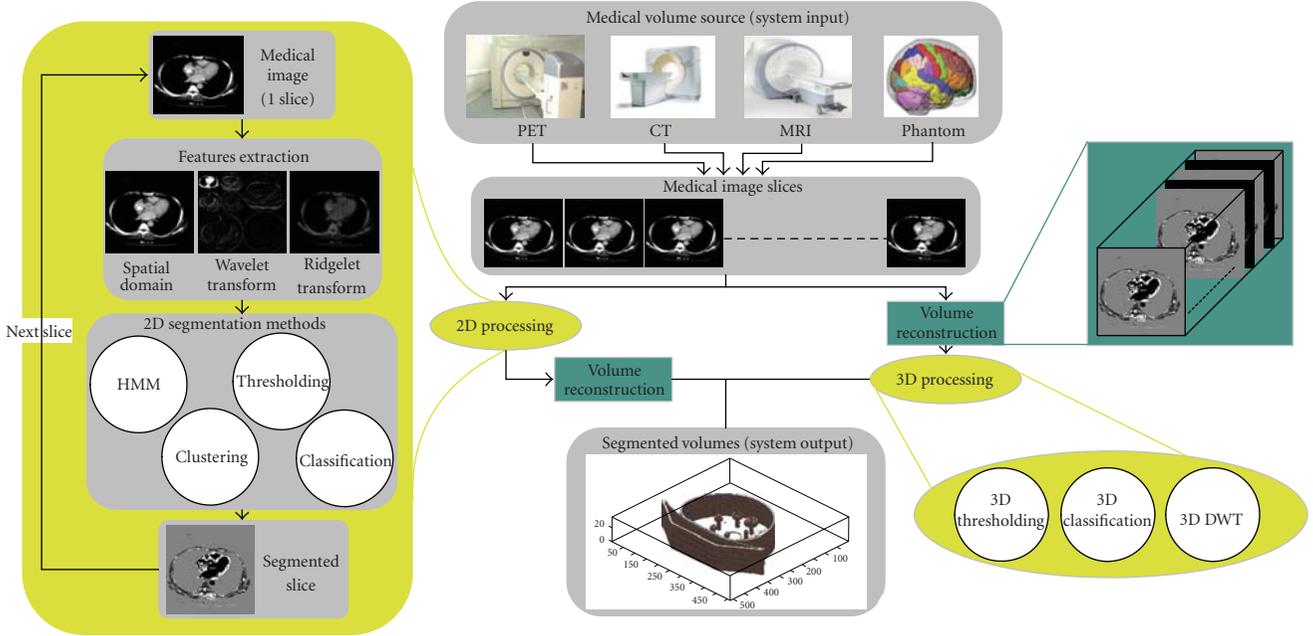


FIGURE 1: Proposed segmentation system.



FIGURE 2: Thresholding technique for medical image segmentation at threshold value $t = 180$. (a) Original image, and (b) Hard thresholding, (c) Soft thresholding.

```

if Pixelvalue  $\leq$  Thresholdvalue then
    Pixelvalue  $\leftarrow$  0
end if
    
```

ALGORITHM 1: pseudo code for hard thresholding.

```

if Pixelvalue  $\leq$  Thresholdvalue then
    Pixelvalue  $\leftarrow$  0
else
    Pixelvalue  $\leftarrow$  Pixelvalue - Thresholdvalue
else if
    
```

ALGORITHM 2: Pseudo code for soft thresholding.

a specific part of an image. Each class is highlighted in the segmented image to illustrate the image as a number of separated regions, and the ROI may be one of those regions. Clustering technique is based on multithreshold values which can be set depending on the image histogram. Amira et al. used in [11] K -means method for segmenting medical volumes; it is the most commonly used clustering technique. K -means will be used in the fourth

section of this research paper as a previously available work to validate the proposed techniques in the comparison tables.

K -means clustering classifies n voxels into K clusters or classes (K less than n). This algorithm chooses the number of clusters (K) then randomly generates K clusters and determine the cluster centers. The next step is assigning each point in the volume to the nearest cluster center and

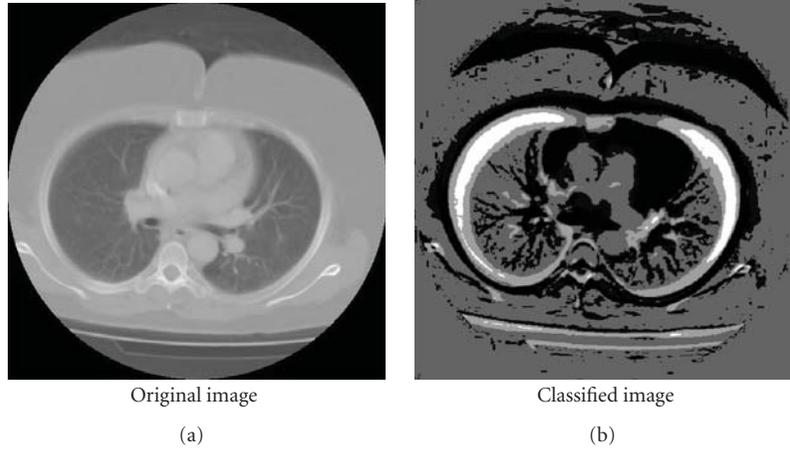


FIGURE 3: Image segmentation using classification technique.

```

(1) Load 3D data set into V
(2) [x y z] = size(V)
    % apply thresholding process for each pixel in the volume
(3) for i = 1 to x do
(4)   for j = 1 to y do
(5)     for k = 1 to z do
(6)       if Pixelvalue ≤ Thresholdvalue then
(7)         Pixelvalue ← 0
(7)       end if
(9)     end for
(10)  end for
(11) end for

```

ALGORITHM 3: pseudo code for 3D-thresholding.

finally recompute the new cluster centers. The two previous steps are repeated until the minimum variance criterion is achieved. This approach is similar to the expectation-maximization algorithm for Gaussian mixture in which they both attempt to find the centers of clusters in the volume. Its main objective is to achieve a minimum intracluster variance V :

$$V = \sum_{i=1}^K \sum_{x_i \in S_i} (x_j - \mu_i)^2, \quad (1)$$

where K is the number of clusters, $S = 1, 2, \dots, K$, and μ_i is the mean of all voxels in cluster i . Figure 3 illustrates an example of segmenting a real CT slice by classifying the image into 6 different classes.

3.3. Wavelet Transform. In the last decade, wavelet transform has been recognized as a powerful tool in a wide range of applications, including image/video processing, numerical analysis, and telecommunication. The advantage of wavelet over existing transforms such as Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) is that wavelet performs a multiresolution analysis of a

signal with localization in both time and frequency [14, 15]. In addition to this, functions with discontinuities and functions with sharp spikes require fewer wavelet basis vectors in the wavelet domain than sine-cosine basis vectors to achieve a comparable approximation. Wavelet operates by convolving the target function with wavelet kernels to obtain wavelet coefficients representing the contributions in the function at different scales and orientations. Wavelet or multiresolution theory can be used alongside segmentation approaches, creating new systems which can provide a segmentation of superior quality to those segmentation approaches computed exclusively within the spatial domain [16].

3.3.1. Discrete Wavelet Transform. Discrete wavelet transform (DWT) can be implemented as a set of filter banks, comprising a high-pass and low-pass filters. In standard wavelet decomposition, the output from the low-pass filter can then be decomposed further, with the process continuing recursively in this manner. According to [17], DWT can be mathematically expressed by(2)

$$\begin{aligned}
 a^j(n) &= \sum_{i=0}^{L-1} l(i) \cdot a^{j-1}(2n-i), \quad 0 \leq n < N_j, \\
 d^j(n) &= \sum_{i=0}^{L-1} h(i) \cdot d^{j-1}(2n-i), \quad 0 \leq n < N_j.
 \end{aligned} \quad (2)$$

The coefficients $a^j(n)$ and $d^j(n)$ refer to approximation and detailed components in the signal at decomposition level j , respectively. The $l(i)$ and $h(i)$ represent the coefficients of low-pass and high-pass filters, respectively.

DWT decomposes the signal into a set of resolution-related views. The wavelet decomposition of an image creates at each scale j a set of coefficient values w_j , with an overall mean of zero. This set of coefficient values w_j contains the same number of voxels as the original 3D volume, and therefore, this wavelet transform is redundant [18, 19]. A nondecimated or redundant wavelet transform is useful for

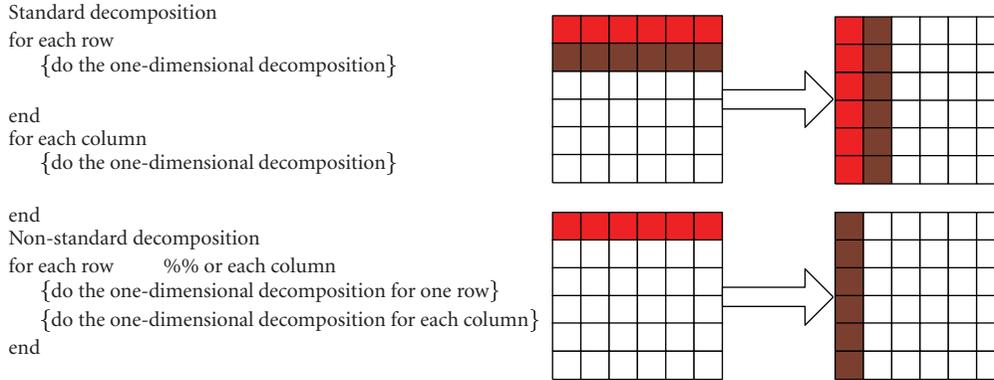


FIGURE 4: Standard and non-standard 2D wavelet transform.

the detection of fine features within the signal. For the case of images, the one-dimensional DWT can be readily extended to two-dimensions. In standard two dimensional wavelet decomposition, the image rows are fully decomposed, with the output being fully decomposed columnwise. In nonstandard wavelet decomposition, all the rows are decomposed by one decomposition level followed by one decomposition level of the columns. Figure 4 illustrates the process of extending 1D-DWT into 2D-DWT.

3.3.2. Discrete Wavelet Packet Transform. Wavelet Packet (WP) is a wavelet transform where the signal is passed through more filters compared to DWT-based approach. Applying DWT or WP on images generates four coefficients; three of them are the detail coefficients, and the remaining one is the average coefficient. It is worth mentioning that the first level of decomposition is the same for both DWT and WP, as illustrated in Figure 5. The differences start being noticed from the second level of decomposition.

The differences between DWT and WP can be seen in the detail coefficients where the next decomposition of DWT is applied on the average coefficients from the previous decomposition (Figure 5(a)). The next decomposition of WP is applied on all previous decomposition coefficients (Figure 5(b)). An example of applying DWT and WP on a phantom slice at different levels of decomposition is illustrated in Figure 6 where the average quadrants in both DWT and WP are the same, but the details quadrants are transformed in WP, unlike in DWT [20]. The number of quadrants in DWT is increased linearly by 3 as the decomposition level is increased by 1 (replace the LL-filter output by its four transforms), as illustrated in

$$\text{Quadrants No. at level } K = (\text{Quadrants No. at level } K - 1) + 3. \tag{3}$$

Details quadrants from previous decomposition levels of WP are transformed; the number of quadrants are increased exponentially by 4 as the level is increased by 1 (replace each filter output by its 4 transforms), as illustrated in

$$\text{Quadrants No. at level } K = (\text{Quadrants No. at level } K - 1) \times 4. \tag{4}$$

3.3.3. 3D Discrete Wavelet Transform (3D-DWT). Section 3.3.1 has demonstrated that 2D-DWT is a generalization of 1D-DWT applied on all rows and columns using either standard or non-standard decomposition. Applying 3D-DWT is not easy; the difference between 2D images and 3D volumes is the third dimension (depth or Z-axis). The expected transform after applying 3D-DWT is illustrated in Figure 7.

The original volume is transformed into 8 octants (features) in the wavelet domain. Mathematically, 3D-DWT is the process of applying 1D-DWT on each vector in Z-axis which has the same X-axis and Y-axis coordinates after applying 2D-DWT for all comprising frames. Algorithm 4 explains the pseudo code for applying 3D-DWT on 3D data set, and the filter structure of 3D Haar wavelet transform is illustrated in Figure 8.

3.4. Ridgelet Transform. Recently, ridgelet transform [21–23] has been generating a lot of interest due to its superior performance over wavelets. While wavelets have been very successful in applications such as denoising and compact approximations of images containing zero dimensional (point singularities), they do not isolate the smoothness along edges that occurs in images [24]. Wavelets are thus more appropriate for the reconstruction of sharp point-like singularities than lines or edges. These shortcomings of wavelets are well addressed by the ridgelet transform, as they extend the functionality of wavelets to higher dimensional singularities, and are effective tools to perform sparse directional analysis [25]. The basic building block of these transforms is the finite radon transform (FRAT), and HWT has been used to perform FRIT. Applying FRAT on image can be presented as a set of projections of the image taken at different angles to map the image space to projection space. Its computation is important in image processing and computer vision for problems such as pattern recognition and the reconstruction of medical images.

For discrete image data, a projection is computed by summation of all data points that lie within specified unit-width strips; those lines are defined in a finite geometry [26]. It can be obtained by applying 1D Inverse Fast Fourier Transform (1D-IFFT) on the 2D Fast Fourier Transform (2D-FFT) restricted to radial lines going through the origin.

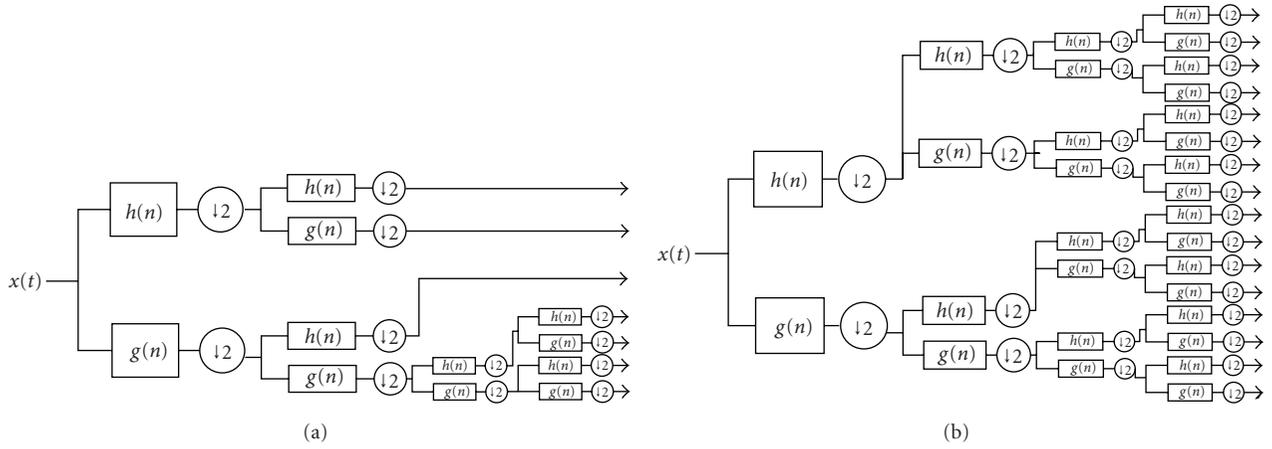


FIGURE 5: Architecture of Haar filters. (a) DWT, (b) WP.

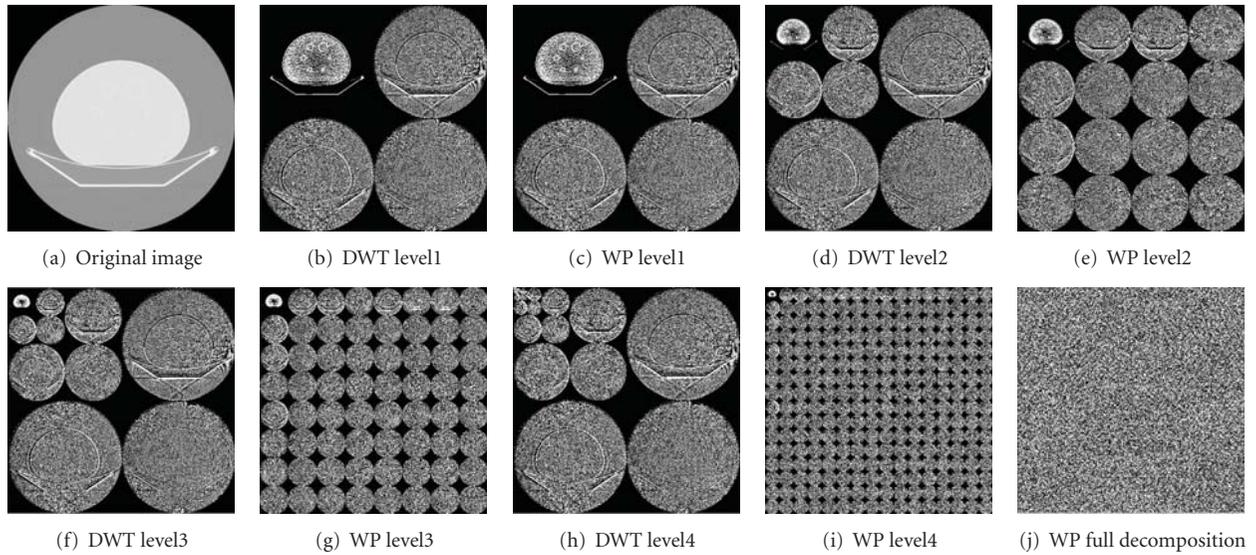


FIGURE 6: DWT and WP for a phantom slice.

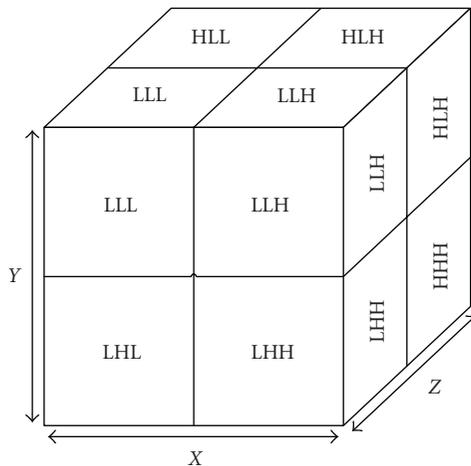


FIGURE 7: 3D volume in wavelet domain.

FRAT of a real function on the finite grid Zp^2 is defined in (5), [16]:

$$r_k[l] = \text{FRAT}_f(k, l) = \frac{1}{\sqrt{P}} \sum_{(i, j) \in L(k, l)} f(i, j). \quad (5)$$

Here, $L(k, l)$ denotes the set of points that make up a line on the lattice Zp^2 as in (6):

$$L(k, l) = \{(i, j) : j = k_i + l(\text{mod } p), i \in Z_p\}, \quad 0 \leq k < p, \\ L(p, l) = \{(l, j) : j \in Z_p\}. \quad (6)$$

To compute the K th radon projection (i.e., the K th row in the array), all pixels of the original image need to be passed once and use P histogrammers: one for every pixel in the row

```

(1) Load 3D data set into V
(2) [x y z] = size(V)
(3) for k = 1 to z do
(4)   apply 2D-DWT for each plane in Z-axis
(5) end for
(6) for i = 1 to x do
(7)   for j = 1 to y do
(7)     apply 1D-DWT for all vectors corresponding to each
         pixel in XY plane
(10)   end for
(11) end for
    
```

ALGORITHM 4: Pseudo code for 3D-DWT.

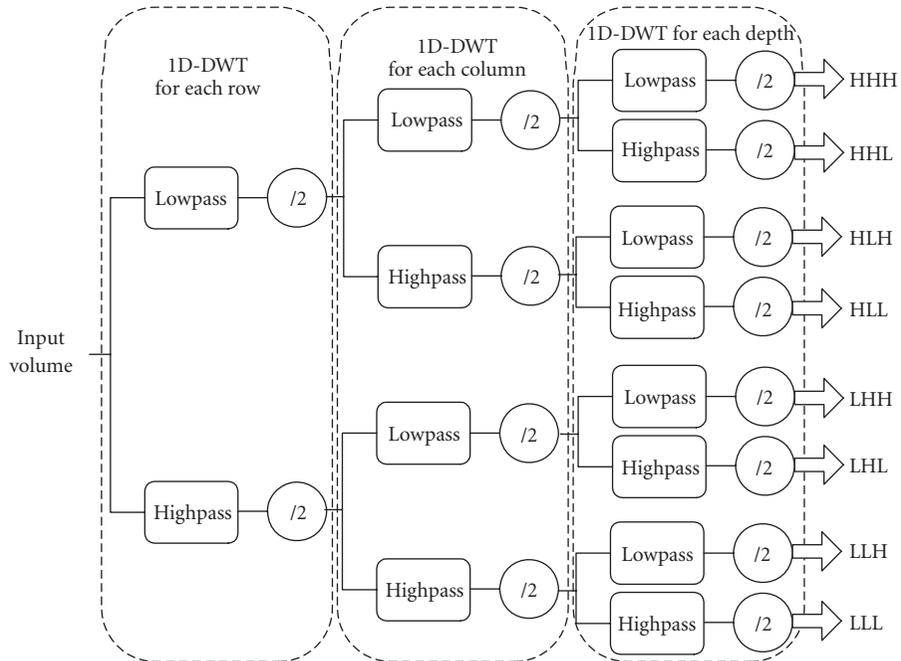


FIGURE 8: Haar filter architecture for 3D-DWT.



FIGURE 9: FRIT block diagram.

[16]. At the end, all P histogrammed values are divided by K to get the average values.

Once the wavelet and radon transforms have been implemented, the ridgelet transform is straightforward. Each output of the radon projection is simply passed through the wavelet transform before it reaches the output multiplier.

As shown in Figure 9, ridgelets use FRAT as a basic building block. FRAT maps a line singularity into point singularity, and the wavelet transform has used to effectively handle the point singularity for the radon domain.

Analysing an object with curve singularity implies that ridgelet coefficient will be not sparse and object with curved singularities is still curved or linear after the radon transform where the wavelet transform cannot detect it properly because it is still not a point singularity [28]. Figure 10 shows a real chest slice from a CT scanner [29] in ridgelet domain at different block sizes.

3.5. Statistical Modeling Using Hidden Markov Models. For block-based segmentation using statistical classification, an image is divided into blocks, and a feature vector is formed for each block by grouping statistics of its pixel intensities [3, 30]. Conventional block-based segmentation algorithms classify each block separately, assuming the independence of feature vectors.

Segmentation in MRFM is achieved by maximizing a posteriori probability of the segmentation depending on

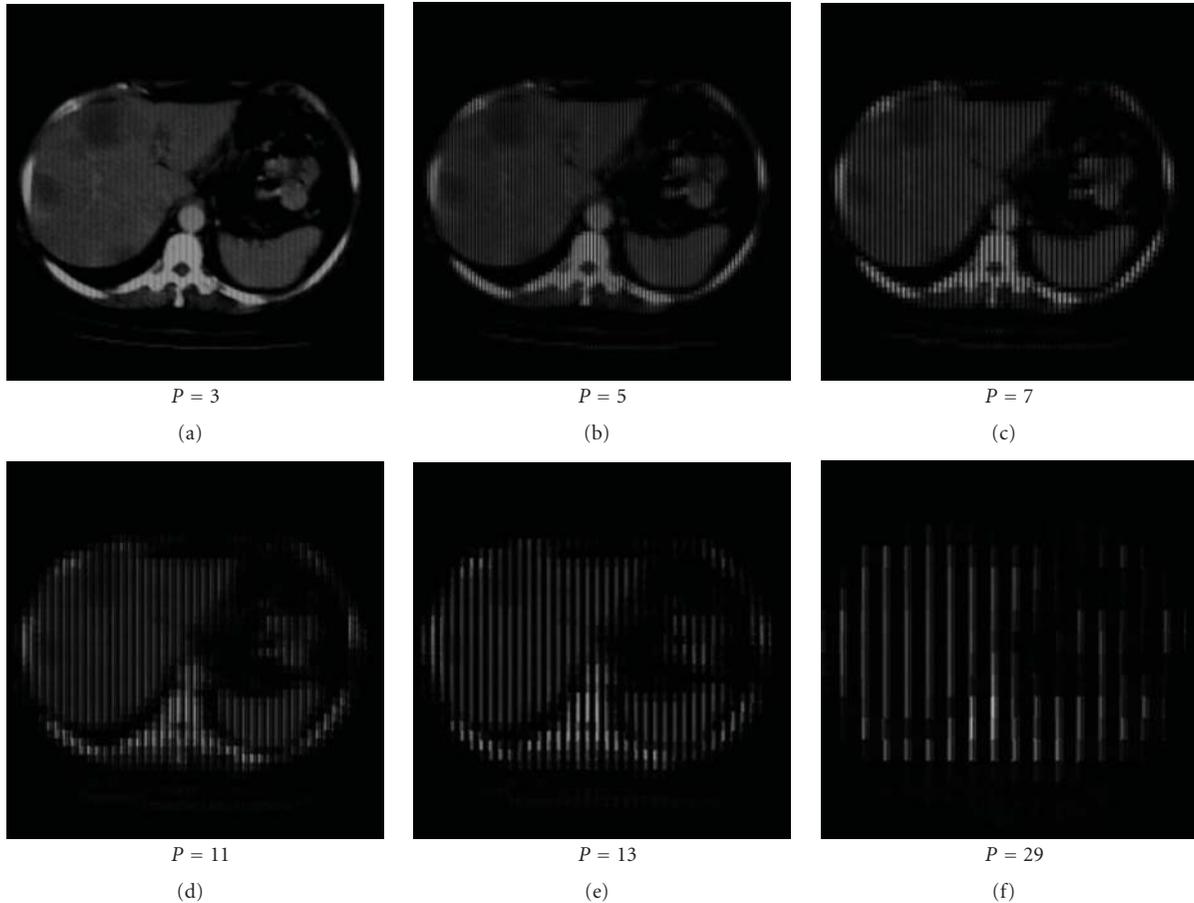


FIGURE 10: Ridgelet transform for real CT image at different block sizes.

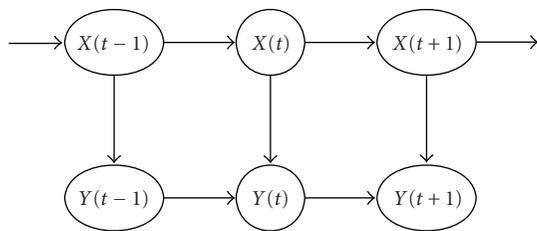


FIGURE 11: Markov chain.

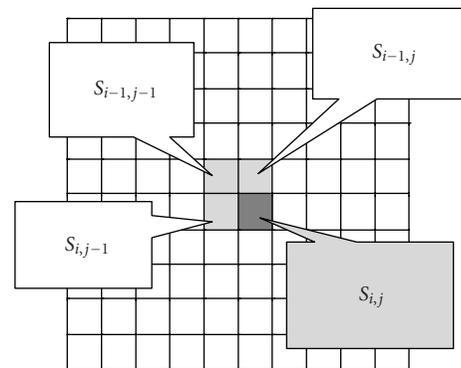


FIGURE 12: Pseudo 2D HMM.

a given image data [31]. MRFM is an earlier version of the HMM, where states in MRFM are directly visible to the observer, and then the state transition probabilities are the only parameters [32]. Markov chain is an edge labeled directed graph (Figure 11), where each node represents a state, and the edge-label have probabilities of moving the state to the end of the directed edge [9, 33].

HMMs represent a widespread approach to the modeling of sequences as they attempt to capture the underlying structure of a set symbol strings. The use of HMM for shape recognition has not been widely addressed. Only a few works have been found to have some similarities with the proposed approach. In the first, He and Kundu [34] utilized

HMMs to model shape contours through autoregressive (AR) coefficients. The use of circular HMM for shape recognition improving scaling and deformation robustness is proposed at [35],[36].

HMM is basically a stochastic finite state automaton, formally defined by the following elements [37]: a set of states $S = S_1, \dots, S_i, \dots, S_j, \dots, S_N$; a state transition probability distribution matrix $A = \{a_{i,j}\} 1 \leq i, j \leq N$ representing the probability to go from state S_i to S_j ; a set

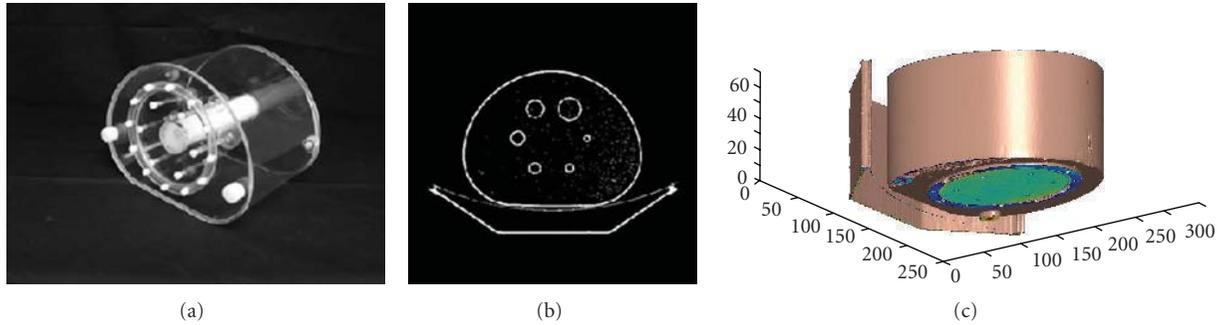


FIGURE 13: (a) NEMA IEC body phantom (DATA SET 1), (b) 2D slice from DATA SET 1, (c) DATA SET 1 after stacking all slices using [27].

of observation symbols $V = \{v_1, v_2, \dots, v_k, \dots, v_M\}$, where v_k used to be a d -dimensional vector (in the case of discrete HMM); an observation symbol probability distribution or emission matrix $B = \{b_j(v_k), 1 \leq k \leq M\} 1 \leq j \leq N$, indicating the probability of emission of symbol v_k when system state is S_j ; an initial state probability distribution $\pi = \{\pi_i\} 1 \leq i \leq N$, representing probabilities of initial states. For convenience, HMM has been denoted as a triplet $\lambda = \{A, B, \pi\}$, which uniquely determines the model.

When modeling a sequence of observation symbols it is usual to use a so-called “left to right HMM” [37], which has only partial state transition matrix, such that $a_{i,j} = 0, j < i, j > i+step$, where *step* is a constant usually equal to 1 or 2.

3.5.1. 2D-HMM. The problem with 2D-HMM is the double dependency of $S_{i,j}$ on its two neighbors, $S_{i-1,j}$ and $S_{i,j-1}$, as illustrated in Figure 12. This does not allow the factorization of computation as in 1D, where $S_{i,j}$ must only depend on one neighbor at a time. However, this neighbor may be the horizontal ($S_{i,j-1}$) or the vertical ($S_{i-1,j}$) [9, 38].

Each slice is a two-dimensional matrix which can be classified by an optimum set of states with maximum probability; these states are mapped into classes or segmented objects. The basic assumption of applying HMM on medical images is to use the embedded-HMM by defining a set of Markovian superstates, within each superstate there is a set of simple Markovian states. The superstate is first chosen using a first-order Markov state transition probability based on the previous superstate. A simple Markov chain is then used to generate observations in this superstate. Thus, superstates are related to rows (or any equal size blocks), and simple states are related to columns (or smaller blocks comprised the superstate).

3.5.2. Defining the Initial States. To generate an observation sequence using HMM, an initial state must be chosen according to the initial state distribution; then an observation sequence should be chosen according to the probability distribution in the initial state [9]. Many feature selection techniques have been tested in medical images, and the best results were achieved using the grey-scale values of the medical image pixels to generate the probability distribution matrix. After defining the initial states, transition to a

new state is taking place according to the state transition probability matrix for the current state and depending on the sequence of observations.

3.5.3. Training HMM. HMM considers observations statistically dependent on neighboring observations through transition probabilities organized in a Markov mesh. Training HMM for images is achieved by dividing the image into nonoverlapping, equally sized blocks, from each of which a feature vector is extracted. Each block and its feature vector evaluate the observation which has its own transition probability matrix. Training HMM produces an estimated state transition probability matrix and estimated emission probability matrix. After building the observation sequence, the model parameters are estimated based on the blocks’ statistics. These classes or states were determined using Viterbi algorithm, which depends on (1) the sequence of observations; (2) estimated state transition probability matrix; (3) emission probability matrix [30, 40]. Finally, pixels which belong to the same class are grouped together to evaluate a segmented image.

3.5.4. Testing HMM. The feature vectors for a testing image are generated to find the set of classes with maximum posteriori according to the trained HMM. The feature vector for each block may be changed at every single state. Once the block state is known, the feature vector will be independent of the other blocks; any two blocks may be more likely to be in the same state if they have close intensities [8]. In other words, testing HMM will generate the state transition probability matrix and emission probability matrix for a given comparable data [8, 9].

4. Results and Analysis

The proposed approach has been tested on NEMA IEC body phantom [41] (DATA SET 1) and real chest images from a CT scanner [29] (DATA SET 2). DATA SET 1 consists of an elliptical water filled cavity with six spherical inserts suspended by plastic rods of inner diameters: 10, 13, 17, 22, 28, and 37 mm [41]; this phantom is illustrated in Figure 13(a). Figure 13(b) is a slice example of the CT outputs which stacked with all other slices to form the 3D

TABLE 1: Wavelet and ridgelet comparisons depending on SNR and processing time.

Domain	Wavelet				Ridgelet		Satial
	Level 1	Level 2	Level 3	$P = 5$	$P = 11$	$P = 31$	
SNR (dB)	10.63	11.14	10.95	10.37	11.43	11.88	7.17
Time (sec)	0.23	0.24	0.50	71.5	29.91	10.01	1.18

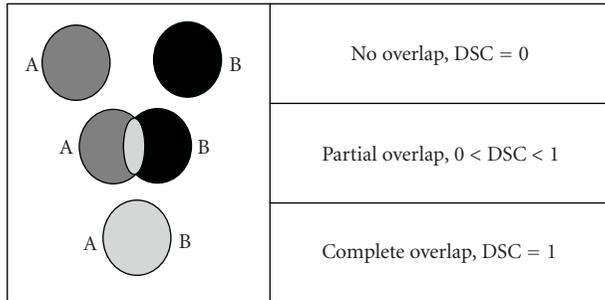


FIGURE 14: DSC overlapping.

volume which is illustrated in Figure 13(c). DATA SET 2 is a CT scanner output of the chest area in a human body [29]. This DATA SET includes blood, bones, muscles, tumor and, other organs in the body.

4.1. Segmentation Performance Metrics. Many techniques can be used for segmentation, and each technique has a different segmentation performance and quality. Listed below are some performance measurement methods that have been used in this paper to test and compare the segmentation techniques.

(i) Dice Similarity Coefficients. Dice Similarity Coefficients (DSCs) are a statistical validation metric used to evaluate the performance of both the reproducibility of manual segmentations and the spatial overlap accuracy of automated probabilistic fractional segmentation. The DSC value is a simple and useful summary measure of spatial overlap, which can be applied to studies of reproducibility and accuracy in image segmentation.

The value of a DSC ranges from 0 indicating no spatial overlap between two sets of binary segmentation results to 1 indicating complete overlap [42]. As illustrated in Figure 14, DSC measures the spatial overlap between two samples, A and B target regions, and is defined as $DSC(A, B) = 2(A \cap B)/(A + B)$ where \cap is the intersection.

(ii) Euclidean Distance. Euclidean Distance (ED) is the straight line distance between two points. It can be used with DATA SET 1 to compare the measured diameters with the original diameters provided with the phantom description [41].

(iii) Processing Time. The future work of this paper is to segment the medical images automatically in real time and get the results while the patient is waiting. the processing time of the segmentation methods are different, and the

processing time may be used as a comparison factor for these methods.

Signal to Noise Ratio (SNR) is also used to differentiate between wavelet and ridgelet output quality. SNR is used in image processing as a physical measure of the sensitivity of an imaging system. Industry standards measure SNR in decibels (dB) of power, and therefore apply the 20 log rule to the peak SNR ratio. Industry standards measure and define sensitivity in terms of the ISO film speed equivalent; image quality is excellent when SNR is 32 dB and if SNR is 20 dB then, it means acceptable image quality.

4.2. Wavelet Versus Ridgelet. Parameters in the wavelet transform are points (x, y) in the cartesian grid. As illustrated in the first part of Figure 15, each point performs a pixel in the image or an entry in a 2D matrix. However, in ridgelet transform, straight lines evaluate the image in the frequency domain rather than those points in wavelet domain. Parameters in ridgelet domain are (β, θ) in a polar domain (second part of Figure 15) where β is the intercept, and θ is the angle [24, 25].

Table 1 illustrates the SNR values of extracted features from DATA SET 2 in spatial domain, wavelet domain at different levels of decomposition, and in ridgelet domain at different block sizes.

It can be seen from Table 1 that small values of SNR have been obtained for all techniques; this is due to the noise from the acquisition systems which will be a part of the medical image itself after the reconstruction of all the slices. Relatively, better SNR values can be achieved with the second level of wavelet decomposition and as the block size (p) is getting bigger with the ridgelet transform where the transformed image is getting more similar to the original image.

4.3. 3D DWT. Applying segmentation techniques on 2D slices requires more time compared to the 3D volumes-based approaches time. The time required to look for the best slice that includes the spheres in full diameters is not required in 3D volume segmentation processes. Segmented volume for DATA SET 1 using 3D-thresholding is illustrated in Figure 16(a), 3D-wavelet technique is applied on the same phantom data (DATA SET 1), and the selected spheres are detected as illustrated in Figure 16(b). Another example of applying 3D-DWT on real CT images for DATA SET 2 is illustrated in Figure 16(c). Table 2 compares the errors in spheres diameters using 3D segmentation techniques with existing measurements using 2D segmentation techniques for slice number 19. ED has been used to measure the spheres diameters and calculate the error percentages for each

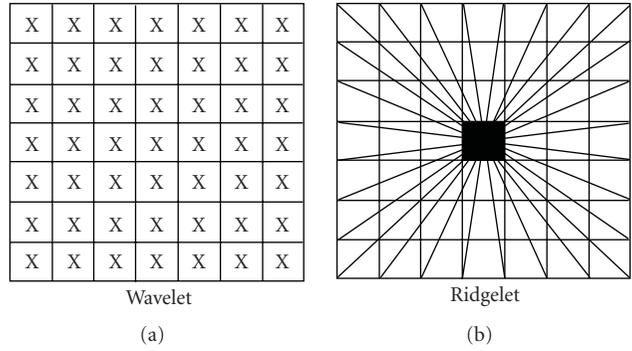


FIGURE 15: Wavelet and ridgelet parameters.

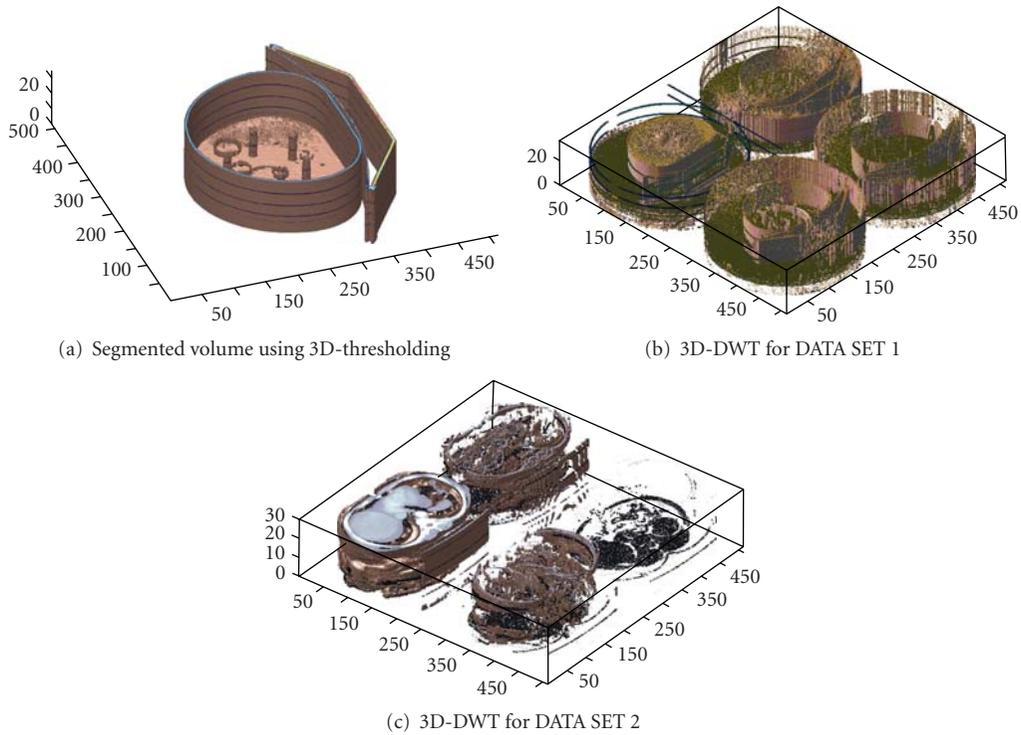


FIGURE 16: 3D segmentation techniques.

TABLE 2: Error percentage of spheres measurement using different segmentation techniques.

Spheres Diameter (mm)			10	13	17	22	28	37
K-means [11]			-13.6	-11.5	-5.77	-5.51	-5.1	-5.01
MRFM [11]			-7.41	-8.69	4.28	4.06	3.9	3.89
Clustering [11]			18.6	16	9	7.5	5.5	1.1
Iterative Thresholding [39]			3	3.1	0.6	0.9	1.1	1.8
2D Thresholding			-4.8	-8.15	0.06	-0.36	1.07	1.14
2D-DWT (Proposed)	Haar	Level 1	-2.9	-2.46	1.35	0.82	0.29	0.05
		Level 2	-10.9	-6.67	3.88	-1.3	-0.76	-1.95
		Level 3	NA*	NA*	5.65	-18.2	2.57	-3.24
Daubechies	Level 1	-7.43	-2.69	0.12	2	2.17	1.81	
	Level 2	-5.2	0.15	-4.24	0.73	0.62	-0.11	
3D Thresholding (Proposed)			0.59	0.77	0.17	1.11	2.2	6.08
3D-DWT (Proposed)			-2.67	-1.93	-0.74	4.75	3.37	0.77

TABLE 3: Performance of using HMMs and MRA for medical image segmentation.

diameters (mm)	10	13	17	22	28	37	Processing time (sec)
HMM (<i>Spatial</i>)	1.5	0.8	0.1	0.8	0.04	1	2007.3
HMM (<i>Wavelet</i>)	NA*	11.07	7.82	2.98	1.09	0.27	1321.1
HMM (<i>Ridgelet</i>)	NA*	NA*	NA*	NA*	NA*	2.41	5757.6

*Diameters could not be detected using this Technique.

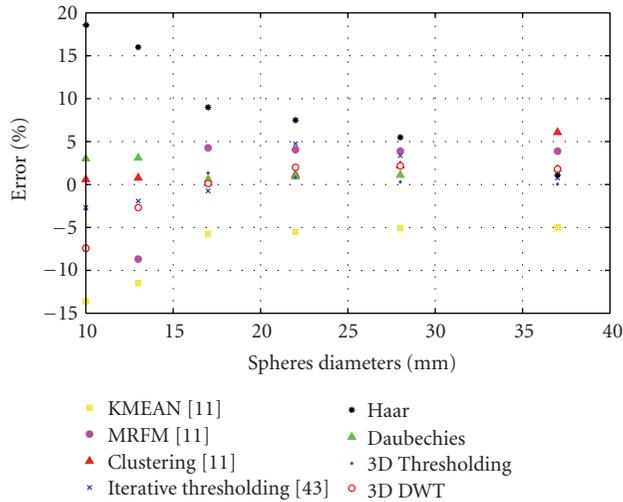


FIGURE 17: Visual comparison for error percentages.

TABLE 4: HMM performance depending on DSC.

Type	joint pixels	No. of pixels	DSC value
Spatial	212521	524288	0.8107
Ridgelet	158905	524288	0.6062

technique sphere diameter error percentages have been calculated according to the following:

$$\text{error\%} = \frac{\text{Measured Diameter} - \text{Actual Diameter}}{\text{Actual Diameter}} \times 100\%. \quad (7)$$

From Table 2, in the case of K -means clustering, tumor volumes are underestimated by approximately 5-6% in most cases, however, for the two smaller spherical inserts, with diameter of 10 mm and 13 mm, respectively, these underestimations are significantly greater. For the smallest sphere, more than a 13% volume discrepancy is recorded, with the K -means algorithm finding it difficult to quantify the tumor accurately. Sphere 2 similarly is massively underestimated (11 : 5%). Unlike K -means clustering, MRFM tends to overestimate the volumes of the spherical inserts, with the exception of Spheres 1 and 2.

Outer diameters have been measured in the case of 3D segmentation, and the inner diameters errors have been calculated based on the thickness of spheres edges. All spheres diameters detected using 3D-thresholding and the errors were over estimated by (0.5–2.5%), they were increasing while the sphere diameter increasing.

Spheres diameters are reduced to the half with each decomposition level of wavelet transform. Three decomposition levels of DWT have been applied on NEMA phantom using two different filters (Haar, Daubechies), and the measured diameters were doubled at each level to produce a fair comparison with the other available techniques. It can be seen that most of the error percentages were decreasing while the spheres diameter increasing, it is worth mentioning that there is no upper bound of the spheres diameters to keep the errors decreasing because the ROI becomes clearer and easier to be detected and measured properly. But tumors in real life are usually very small in the early stage cancer, and the problem is to detect those turnouts as soon as possible.

By applying one decomposition level of 3D-DWT on spatial domain and using the LLL filter output, underestimated percentages have been achieved for the three small spheres (10, 13, and 17 mm) and overestimated percentages for the three big spheres (22, 28, and 37 mm). DWT proved efficient in detecting the big obstacles where the biggest sphere (37 mm) was detected with a very small error percentage (<0.8%).

The two smallest spherical inserts are still underestimated in all techniques except the 3D-thresholding. The large volumetric errors encountered using this acquisition exist as a consequence of the poor slice thickness setting selected for the scan. The 4.25 mm slice thickness causes large fluctuations in transaxial tumor areas to occur between image slices. This problematic characteristic occurs most notably with the smallest spherical inserts, where single voxel reallocation causes a large deviation in percentage error. In Figure 16, the percentage error computed between the actual sphere volume and the volumes obtained using all methodologies for each of the six tumor inserts is plotted. It can be seen that all techniques are settled down according to the error percentages as the sphere diameters increased (Figure 17).

4.4. HMM Experimental Results. HMMs have been used for segmentation, which can be applied either in the spatial or multiresolution domain using wavelet or ridgelet transforms. The weakness of HMMs is its long processing time, compared with the other evaluated techniques.

DATA SET 1 has been utilized to perform the experimental study using HMMs and MRA for medical image segmentation, and the achieved results are illustrated in Table 3. Figures 18 and 19 illustrate the outputs of applying HMM on a slice from DATA SET 1 and real brain slice, respectively.

From Table 3, applying HMMs on NEMA phantom slice in ridgelet domain failed to segment ROI where most of

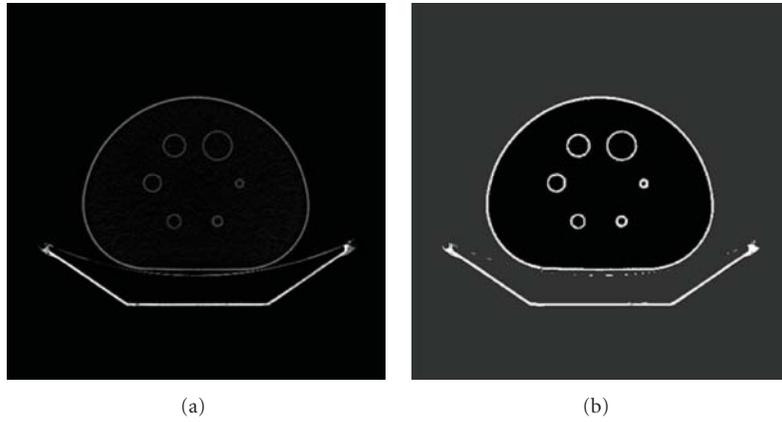


FIGURE 18: Segmented image using thresholding technique (a) and HMM (b).

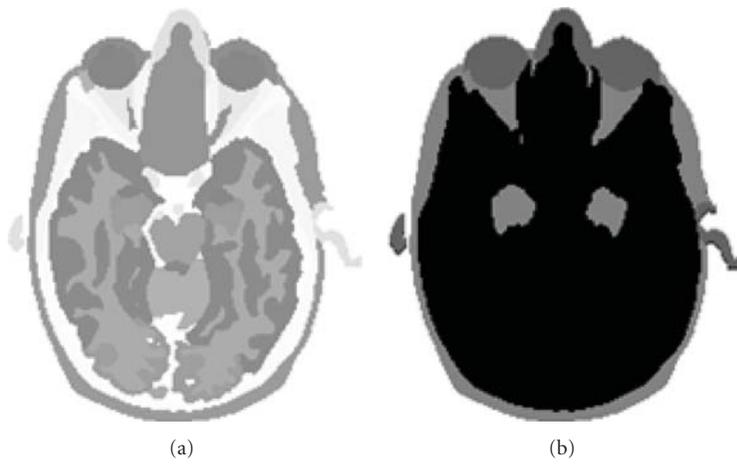


FIGURE 19: Image Segmentation using HMM. Original (a) and HMM (b)

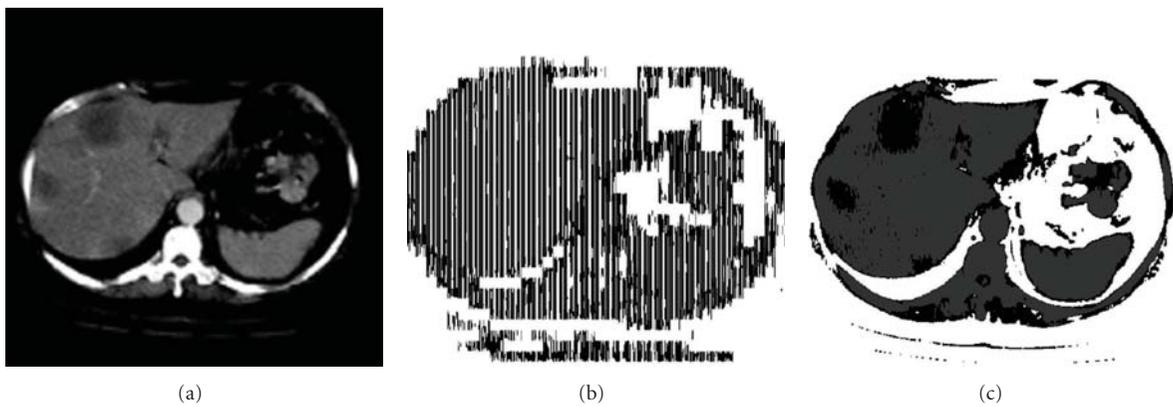


FIGURE 20: Image Segmentation using HMM. Original (a), HMM on ridgelet (b), and HMM on spatial (c).

the spheres diameters could not be detected. As well as the long computational time problem, it is five times more than the required time for applying HMMs on the slice in the wavelet domain. Transforming any image into ridgelet domain changes the comprising pixel specifications as well as the dimensions of the image. The padding row is added

to each block leading to long computational time for HMM segmentation. Relatively, the most accurate results have been achieved with applying HMMs directly on the spatial domain without any transformation. But in wavelet domain, applying HMMs requires less time which make it very useful in real time segmentation systems.

It can be seen from Figure 18 that different obstacles were merged such as bones and brain; the same thing happened in Figure 19 where the liver is merged with the skin. This merge is due to the number of HMM classes (states) where HMMs decided that brain and bones in Figure 18 will be in the same class because just three HMM classes were used. But each obstacle will be in separate class if the number of classes increased to the number of all obstacles in the image, the same explanation for Figure 19. There are some available techniques that can be used to define the best number of classes in HMM such as BiC.

HMMs have been also applied on DATA SET 2 as illustrated in Figure 20, and the DSC values for the segmented slices have been measured and illustrated in Table 4 which evaluates the performance based on a manual segmented image where better results have been achieved using HMMs in the spatial domain.

Many techniques have been previously implemented for medical volumes segmentation; some of them were illustrated in this paper and compared with the proposed techniques. Promising results have been achieved using 3D segmentation techniques directly to medical volumes and the statistical models using HMMs; both techniques have the same problem which is the computational time. Many acceleration processing methods have been recently implemented such as FPGAs, Matlab accelerators, Feature Reduction (FR) techniques, GPUs and many more. HMM can be applied with different MRA transforms such as wavelet, ridgelet, and curvelet to achieve promising results. It is worth mentioning that based on the experiments carried out on these specific medical data in this paper, HMMs can be classified as one of the ideal medical volume segmentation techniques compared to the other proposed techniques.

5. Conclusions

Segmentation is very important for medical image processing, to detect tumors and regions of interest for radiotherapy planning and cancer diagnosis. A novel and sophisticated segmentation system was developed specifically for 3D data segmentation. The developed techniques within the system have been tested on phantom data [41]. The system used to quantify tumors within the data of predefined volumes, and these results were compared with those obtained from 2D approaches such as thresholding. Thresholding addressed many problems in multiresolution analysis including denoising and edge detection, but data loss was the main problem caused. Statistical models using hidden Markov models have been also investigated for segmentation, which can be deployed in the spatial domain or multiresolution domain. The weakness of HMMs is its long computation time required for the calculation of their models which was much smaller compared with the other evaluated techniques.

The proposed system was also tested on other data set for real human chest images from a CT scanner and has shown promising results. Ongoing research is focusing on the implementation of other 3D novel feature extraction techniques for medical images based on 3D-ridgelet and

3D-curvelet transforms. In order to speed up the proposed techniques; a graphical processing unit (GPU) will be deployed with more focus on the implementation of higher dimensional HMMs for a more accurate and automatic volume segmentation system.

References

- [1] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Transactions on Medical Imaging*, vol. 22, no. 3, pp. 441–458, 2003.
- [2] D. Montgomery, *Multiscale compression and segmentation of volumetric oncological PET imagery*, Ph.D. thesis, 2006.
- [3] D. W. G. Montgomery, A. Amira, and H. Zaidi, "Fully automated segmentation of oncological PET volumes using a combined multiscale and statistical model," *Medical Physics*, vol. 34, no. 2, pp. 722–736, 2007.
- [4] H. Zaidi, M. Diaz-Gomez, A. Boudraa, and D. O. Slosman, "Fuzzy clustering-based segmented attenuation correction in whole-body PET imaging," *Physics in Medicine and Biology*, vol. 47, no. 7, pp. 1143–1160, 2002.
- [5] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2nd edition, 2001.
- [6] A. Jensen and A. la Cour-Harbo, *Ripples in Mathematics: The Discrete Wavelet Transform*, Springer, Berlin, Germany, 2001.
- [7] D. Agrafiotis, D. Bull, and N. Canagarajah, "Three-dimensional coding of volumetric medical images using regions of interest," in *Proceedings of the International Conference on Visual Information Engineering (VIE '03)*, pp. 194–197, July 2003.
- [8] J. Li, A. Najmi, and R. M. Gray, "Image classification by a two-dimensional hidden Markov model," *IEEE Transactions on Signal Processing*, vol. 48, no. 2, pp. 517–533, 2000.
- [9] J. Li and J. Z. Wang, "Studying digital imagery of ancient paintings by mixtures of stochastic models," *IEEE Transactions on Image Processing*, vol. 13, no. 3, pp. 340–353, 2004.
- [10] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 233–260, 1988.
- [11] A. Amira, S. Chandrasekaran, D. W. G. Montgomery, and I. Servan Uzun, "A segmentation concept for positron emission tomography imaging using multiresolution analysis," *Neurocomputing*, vol. 71, no. 10–12, pp. 1954–1965, 2008.
- [12] S. G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1522–1531, 2000.
- [13] X. Zhang and M. Desai, "Wavelet based automatic thresholding for image segmentation," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '97)*, vol. 1, pp. 224–227, 1997.
- [14] I. Daubechies, "Wavelet transforms and orthonormal wavelet bases," in *Different Perspectives on Wavelets (San Antonio, Tex, 1993)*, vol. 47 of *Proceedings of Symposia in Applied Mathematics*, pp. 1–33, American Mathematical Society, Providence, RI, USA, 1993.
- [15] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
- [16] K. Rajpoot and N. Rajpoot, "Hyperspectral colon tissue cell classification," in *Medical Imaging*, Proceedings of SPIE, 2004.

- [17] I. S. Uzun and A. Amira, "Design and FPGA implementation of finite ridgelet transform," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 6, pp. 5826–5829, May 2005.
- [18] E. J. Stollnitz, T. D. DeRose, and D. H. Salestin, "Wavelets for computer graphics: a primer—part 1," *IEEE Computer Graphics and Applications*, vol. 15, no. 3, pp. 76–84, 1995.
- [19] A. Haar, "Zur theorie der orthogonalen funktionensysteme—erste mitteilung," *Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, 1910.
- [20] Z. Shi and R. Shibasaki, "An approach to image segmentation using multiresolution analysis of wavelets," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 6, pp. 810–815, IEEE, Tokyo, Japan, 1999.
- [21] M. N. Do and M. Vetterli, "Orthonormal finite ridgelet transform for image compression," in *Proceedings of the International Conference on Image Processing (ICIP '00)*, vol. 2, pp. 367–370, September 2000.
- [22] E. Candès and D. Donoho, "Curvelets—a surprisingly effective nonadaptive representation for objects with edges," in *Curves and Surfaces*, pp. 105–120, Vanderbilt University Press, 2000.
- [23] E. Candès, *Ridgelets: theory and application*, Ph.D. thesis, Department of Statistics, Stanford University, Stanford, Calif, USA, 1998.
- [24] E. J. Candès and D. L. Donoho, "Ridgelets: a key to higher-dimensional intermittency?" *Philosophical Transactions of the Royal Society A*, vol. 357, no. 1760, pp. 2495–2509, 1999.
- [25] J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.
- [26] J. He, "A characterization of inverse Radon transform on the Laguerre hypergroup," *Journal of Mathematical Analysis and Applications*, vol. 318, no. 1, pp. 387–395, 2006.
- [27] A. Balter, *Dicom to 3 Dimentional*, Pacific Northwest National Laboratory, 2009.
- [28] H. Biao, L. Fang, and J. Licheng, "Linear feature detection based on ridgelet," *Science in China, Series E*, vol. 46, no. 2, 2003.
- [29] *Computed Tomography Scanner*, King Abdullah University Hospital, Ramtha, Jordan, 2009.
- [30] J. Lu and L. Carin, "HMM-based multiresolution image segmentation," in *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, pp. 3357–3360, May 2002.
- [31] A. Huang, R. Abugharbieh, and R. Tam, "Image segmentation using an efficient rotationally invariant 3D region-based hidden Markov model," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '08)*, pp. 107–111, Anchorage, Alaska, USA, 2008.
- [32] D. DeMonthon and M. Vuilleumier, "Image distance using hidden Markov models," in *Proceedings of the 15th International Conference on Pattern Recognition (ICPR '00)*, vol. 3, p. 3147, 2000.
- [33] J. Jiten and B. Merialdo, "Semantic image segmentation with a multidimensional hidden Markov model," in *Proceedings of the 13th International Multimedia Modeling Conference*, vol. 4351 of *Lecture Notes in Computer Science*, pp. 616–624, Springer, Singapore, 2007.
- [34] Y. He and A. Kundu, "2-D shape classification using hidden Markov model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1172–1184, 1991.
- [35] N. Arica and F. T. Yarman Vural, "A shape descriptor based on circular hidden Markov model," in *Proceedings of the 15th IEEE International Conference on Pattern Recognition*, vol. 1, pp. 924–927, 2000.
- [36] J. Solomon, J. A. Butman, and A. Sood, "Segmentation of brain tumors in 4D MR images using the hidden Markov model," *Computer Methods and Programs in Biomedicine*, vol. 84, no. 2-3, pp. 76–85, 2006.
- [37] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [38] X. Ma, D. Schonfeld, and A. Khokhar, "Distributed multi-dimensional hidden Markov model: theory and application in multiple-object trajectory classification and recognition," in *Multimedia Content Access: Algorithms and Systems II*, Proceedings of SPIE, 2008.
- [39] W. Jentzen, L. Freudenberg, E. G. Eising, M. Heinze, W. Brandau, and A. Bockisch, "Segmentation of PET volumes by iterative image thresholding," *Journal of Nuclear Medicine*, vol. 48, no. 1, pp. 108–114, 2007.
- [40] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm," *IEEE Transactions on Medical Imaging*, vol. 20, no. 1, pp. 45–57, 2001.
- [41] International Electrotechnical Commission (IEC), 61675-1, Geneva, Switzerland, 1998, National Electrical Manufacturers Association (NEMA), Standards Publication No. NU2, Washington, DC, USA, 2001.
- [42] K. H. Zou, S. K. Warfield, A. Bharatha et al., "Statistical validation of image segmentation quality based on a spatial overlap, index 1, scientific reports," *Academic Radiology*, vol. 11, no. 2, pp. 178–189, 2004.

Research Article

Unsupervised Topographic Learning for Spatiotemporal Data Mining

Guénaël Cabanes and Younès Bennani

LIPN-CNRS, UMR 7030, Université de Paris 13, 99, avenue J-B. Clément, 93430 Villetaneuse, France

Correspondence should be addressed to Guénaël Cabanes, guenael.cabanes@lipn.univ-paris13.fr

Received 14 June 2010; Revised 5 September 2010; Accepted 7 September 2010

Academic Editor: Abbes Amira

Copyright © 2010 G. Cabanes and Y. Bennani. 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.

In recent years, the size and complexity of datasets have shown an exponential growth. In many application areas, huge amounts of data are generated, explicitly or implicitly containing spatial or spatiotemporal information. However, the ability to analyze these data remains inadequate, and the need for adapted data mining tools becomes a major challenge. In this paper, we propose a new unsupervised algorithm, suitable for the analysis of noisy spatiotemporal Radio Frequency IDentification (RFID) data. Two real applications show that this algorithm is an efficient data-mining tool for behavioral studies based on RFID technology. It allows discovering and comparing stable patterns in an RFID signal and is suitable for continuous learning.

1. Introduction

In recent years, the size of datasets has shown an exponential growth. A study exhibits that the amount of data doubles every three years [1]. In application areas such as robotics, computer vision, mobile computing, and traffic analysis, huge amounts of data are generated and stored in databases, explicitly or implicitly containing spatial or spatiotemporal information. For instance, the proliferation of location-aware devices gives rise to vast amounts of frequently updated telecommunication and traffic data, and satellites generate terabytes of image data daily. These huge collections of spatiotemporal data often hide possibly interesting information and valuable knowledge. It is obvious that a manual analysis of these data is impossible, and data mining might provide useful tools and technology in this setting. Spatiotemporal data mining is an emerging research area that is dedicated to the development of novel algorithms and computational techniques for the successful analysis of large spatiotemporal databases and the disclosure of interesting knowledge in spatiotemporal data. However, the ability to analyze these data remains inadequate and the need for adapted data mining tools becomes a major challenge.

As the study of data streams and large databases is a difficult problem because of the computing costs and the big

storage volumes involved, two issues appear to play a key role in such an analysis: (i) a good condensed description of the data properties [2, 3] and (ii) a measure capable of detecting changes in the data structure [4, 5]. In this paper, we propose a new algorithm which combines two methods [6, 7] to perform these two tasks. The solution we propose consists of an algorithm, which constructs an abstract representation of the datasets, performs automatically a clustering of the data based on their representations, and is able to evaluate the dissimilarity between two distinct datasets. The abstract representation is based on the learning of a variant of Self-Organizing Map (SOM) [8], which is enriched with information extracted from the data. Then, the underlying data density function is estimated from the abstract representation. The dissimilarity is a measure of the divergence between two estimated densities. A great advantage of this method is that each enriched SOM is at the same time a very informative and a highly condensed description of the data distribution that can be stored easily for a future use. Moreover, as the algorithm is effective both in terms of computational complexity and in terms of memory requirements, it can be used for comparing large datasets or for detecting changes in data streams.

In this work, we focus on the analysis of RFID data. RFID is an advanced tracking technology. The RFID tags, which

consist of a microchip and an antenna, must be used with a reader that can detect simultaneously a lot of tags in a single scan. A computer is used to store the data about the position of each tag for each scan in a database. This allows different analyses. RFID, thanks to miniaturization, offers the advantage of automation and overcomes the constraints imposed by video analyzes. The evolution of these data over time and their spatial position require the exploration of multiple data sets described in high dimension spaces.

The proposed algorithm presents some interesting properties to deal with RFID data.

- (i) It allows a compact representation of each trajectory in a linear computational cost. This is important regarding that the total amount of recorded data can be high. Indeed, each tag's position is recorded very frequently (sometime less than each second) during several hours and many tags are followed simultaneously.
- (ii) It is able to deal with noisy data and to find automatically a suitable number of clusters, without constraints about the clusters' shape. Moreover, it presents good clustering performances comparing to traditional algorithms. This is appreciable as RFID trajectories are very noisy in our application and the extraction of general patterns become a difficult task. The algorithm is thus used to perform a clustering of each trajectory from its representation.
- (iii) It is suitable for trajectories comparisons. As new trajectory record can be added to the database at any time, it is important to compare patterns of new trajectories with older ones. The algorithm can evaluate the similarity of two trajectories' representations using information about underlying distribution of the tags' movements. This approach is very resistant to noise and is more reliable than distance-based methods.

Thus, the algorithm combines all needed properties—good abstraction performances, low memory, and computational cost, suitable for noisy experimental data analysis—to be a good candidate for RFID data mining.

The remainder of this paper is organized as follows. Section 2 presents the algorithm. Properties analysis and results are described in Section 3. Section 4 introduces the adaptation of the algorithm to deal with RFID data then describes two real application (mining customers trajectories in a supermarket and analyzing migration behavior of an ant's colony). A conclusion is given in Section 5.

2. Proposed Algorithm

2.1. General Algorithm Schema. The basic assumption in this work is that it is possible to define prototypes in the data space and to calculate a distance measure between data points and prototypes. First, each dataset is modeled using an enriched SOM model, constructing an abstract representation which is supposed to capture the essential properties of the data. Then, a clustering of the data is

computed, in order to catch the global structure of these data. Finally, the density function of each dataset is estimated from the abstract representation and different datasets can be compared using a dissimilarity measure based upon these density functions.

The idea is to combine the dimension reduction and the fast learning capabilities of SOM to construct a new vector space then apply other analysis in this space. These are called two-level methods. The two-level methods are known to reduce greatly the computational time, the effects of noise, and the “curse of dimensionality” [6]. Furthermore, it allows some visual interpretation of the result using the two-dimensional map generated by the SOM.

The algorithm proceeds in three steps.

- (1) The first step is the learning of the enriched SOM. During the learning, each SOM prototype is extended with novel information extracted from the data. These information will be used in the following step to find clusters in the data and to infer the density function. More specifically, the attributes added to each prototype are the following.
 - (a) *Density modes.* It is a measure of the data density surrounding the prototype (local density). The local density is a measure of the amount of data present in an area of the input space. We use a Gaussian kernel estimator [9] for this task.
 - (b) *Local variability.* It is a measure of the data variability that is represented by the prototype. It can be defined as the average distance between the prototypes and the represented data.
 - (c) *The neighborhood.* This is a prototype's neighborhood measure. The neighborhood value of two prototypes is the number of data that are well represented by each one.
- (2) The second step is the clustering of the data using density and connectivity information so as to detect low-density boundary between clusters.
- (3) The third step is the construction, from each cluster (i.e., a set of enriched prototypes in a SOM), of a density function which will be used to estimate the density in the input space. This function is constructed by induction from the information associated to the prototypes of the SOM and is represented as a mixture model of spherical normal functions.
- (4) The last step accomplishes the comparison of two different datasets (e.g., clusters from different databases) using a dissimilarity measure able to compare the two density functions constructed in the previous steps.

2.2. Data Structure Modeling with Enriched SOM. Kohonen SOM can be defined as a competitive unsupervised learning neural network [8]. When an observation is recognized, the activation of an output cell—competition layer—inhibits the activation of other neurons and reinforce itself. It is said that

it follows the so called “Winner Takes All” rule. Actually, neurons are specialized in the recognition of one kind of observation. An SOM consists of a two-dimensional map of neurons which are connected to n -inputs according to n weights connections and to their neighbors with topological links. The training set is used to organize these maps under topological constraints of the input space. Thus, a mapping between the input space and the network space is constructed; two close observations in the input space would activate two close units of the SOM. An optimal spatial organization is determined by the SOM from the input data, and when the dimension of the input space is less than three, both position of weights vectors (also called prototypes) and direct neighborhood relations between cells can be represented visually. Thus, a visual inspection of the map provides qualitative information about the map and the choice of its architecture. The winner neuron updates its prototype vector, making it more sensitive for later presentation of that type of input. This allows different cells to be trained for different types of data. To achieve a topological mapping, the neighbors of the winner neuron can adjust their prototype vector towards the input vector as well, but at a lesser degree, depending on how far away they are from the winner. Usually, a radial symmetric Gaussian neighborhood function K_{ij} is used for this purpose.

In our algorithm, the SOM’s prototypes will be “enriched” by adding new numerical values extracted from the dataset.

The enrichment algorithm proceeds in three phases.

Input:

- (i) the data $X = \{x_k\}_{k=1}^N$.

Output:

- (i) the density D_i and the local variability s_i associated to each prototype w_i ,
- (ii) the neighborhood values $v_{i,j}$ associated with each pair of prototype w_i and w_j .

Algorithm:

(1) Initialization:

- (i) initialize the SOM parameters,
- (ii) For all i, j initialize to zero the local densities (D_i), the neighborhood values ($v_{i,j}$), the local variability (s_i) and the number of data represented by $w_i(N_i)$.

(2) Choose randomly a data $x_k \in X$:

- (i) compute $d(w, x_k)$, the distance between the data x_k and each prototype w_i ,

- (ii) find the two closest prototypes (BMUs: Best Match Units) w_{u^*} and $w_{u^{**}}$

$$\begin{aligned} u^* &= \arg \min_i (d(w_i, x_k)), \\ u^{**} &= \arg \min_{i \neq u^*} (d(w_i, x_k)). \end{aligned} \quad (1)$$

(3) Update structural values:

- (i) number of data: $N_{u^*} = N_{u^*} + 1$,
- (ii) variability: $s_{u^*} = s_{u^*} + d(w_{u^*}, x_k)$,
- (iii) density: For all i , $D_i = D_i + (1/\sqrt{2\pi}h)e^{-d(w_i, x_k)^2/2h^2}$,
- (iv) neighborhood: $v_{u^*, u^{**}} = v_{u^*, u^{**}} + 1$.

(4) Update the SOM prototypes w_i as defined in [8].

(5) Repeat T times step (2) to (4).

(6) Final structural values: For all i , $s_i = s_i/N_i$ and $D_i = D_i/N$.

In this study, we used the default parameters of the SOM Toolbox [10] for the learning of the SOM, and we use $T = \max(N, 50 \times M)$ as in [10]. The number M of prototypes must neither be too small (the SOM does not fit the data well) nor too large (time consuming). If N is not too big, to choose M close to \sqrt{N} seems to be a good trade off [10]. The last parameter to choose is the bandwidth h . The choice of h is important for good results, but its optimal value is difficult to calculate and time consuming (see [11]). A heuristic that seems relevant and gives good results consists of defining h as the average distance between a prototype and its closest neighbor [6].

At the end of this process, each prototype is associated with a density and a variability value, and each pair of prototypes is associated with a neighborhood value. The substantial information about the distribution of the data is captured by these values. Then, it is no longer necessary to keep data in memory.

2.3. Data Clustering. Various prototypes-based approaches have been proposed to solve the clustering problem [12, 13]. However, the obtained clustering is never optimal, since part of the information contained in the data is not represented by the prototypes. In [6], a new method of prototypes’ clustering is proposed, the Density-based Simultaneous 2-Level—SOM algorithm (DS2L-SOM) that uses density and neighborhood information to optimize the clustering. The main idea is that the core part of a cluster can be defined as a region with high density. Then, in most cases, the cluster borders are defined either by low-density region or “empty” region between clusters (i.e., large intercluster distances) [12].

Here, DS2L-SOM uses information learned by the enriched SOM. Figure 1 shows an example of the different stages of the clustering algorithm: prototypes have been learned with the enriched SOM algorithm; they are represented by hexagons with preservation of the neighborhood.

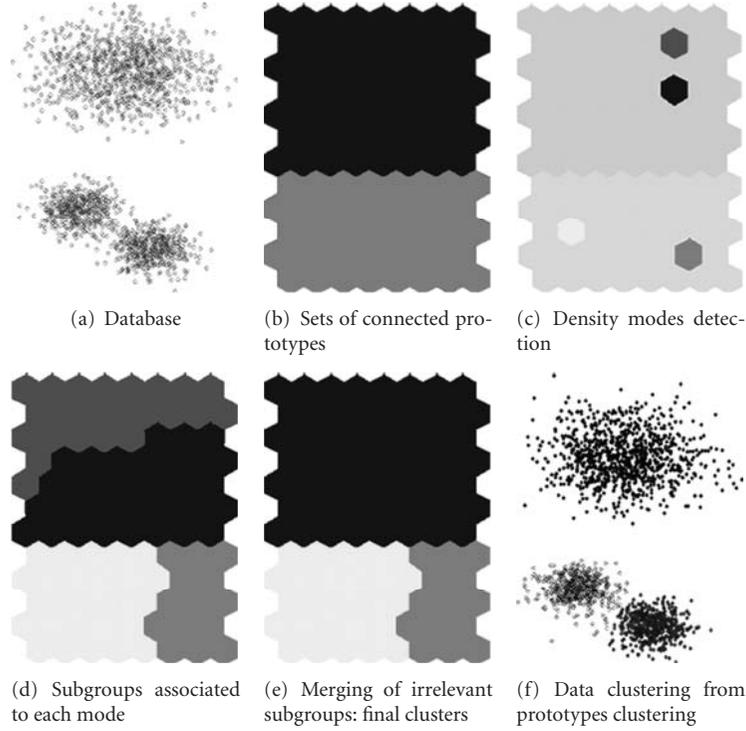


FIGURE 1: Example of a sequence of the different stages of the clustering algorithm.

At the end of the enrichment process (Section 2.2), each set of prototypes linked together by a neighborhood value $\nu > threshold$ define well-separated clusters. This is useful to detect borders defined by large intercluster distances (Figure 1(b)). The estimation of the local density (D) is used to detect cluster borders defined by low density. Each cluster is defined by a local maximum of density (density mode, Figure 1(c)). Thus, a “Watersheds” method [14] is applied on prototypes’ density for each well-separated cluster to find low-density area inside these clusters, in order to characterize density defined subclusters (Figure 1(d)). For each pair of adjacent subgroups, we use a density-dependent index [15] to check if a low-density area is a reliable indicator of the data structure, or whether it should be regarded as a random fluctuation in the density (Figure 1(e)). This process is very fast because generally the number of prototypes is small. The combined use of these two types of group definition can achieve very good results despite the low number of prototypes in the map and is able to detect automatically the number of cluster.

2.4. Comparisons of Data Distributions. Each dataset is modeled using an enriched Self-Organizing Map (SOM) model, constructing an abstract representation which is supposed to capture the essential data structure. Each of the datasets is partitioned using the DS2L-SOM algorithm. In order to be able to compare different clusters from different databases, the algorithm first estimate the underlying density function of each clusters, then use a dissimilarity measure based upon the density functions for the comparison.

The first objective of this step is to estimate the density function which associates a density value to each point of the input space. An estimation of some values of this function have been calculated (i.e., D_i) at the position of the prototypes representing a cluster. An approximation of the function must now be inferred from these values.

The hypothesis here is that this function may be properly approximated in the form of a mixture of Gaussian kernels. Each kernel K is a Gaussian function centered on a prototype. The density function can; therefore; be written as:

$$f(x) = \sum_{i=1}^M \alpha_i K_i(x), \quad (2)$$

with

$$K_i(x) = \frac{1}{N\sqrt{2\pi}h_i} e^{-d(w_i,x)^2/2h_i^2}. \quad (3)$$

The most popular method to fit mixture models (i.e., to find h_i and α_i) is the expectation-maximization (EM) algorithm [16]. However, this algorithm needs to work in the data input space. As here we work on enriched SOM instead of dataset, we cannot use EM algorithm.

Thus, we propose the heuristic to choose h_i

$$h_i = \frac{\sum_j (v_{i,j}/(N_i + N_j)) (s_i N_i + d_{i,j} N_j)}{\sum_j v_{i,j}}, \quad (4)$$

$d_{i,j}$ is the distance between w_i and w_j . The idea is that h_i is the standard deviation of data represented by K_i . These data are

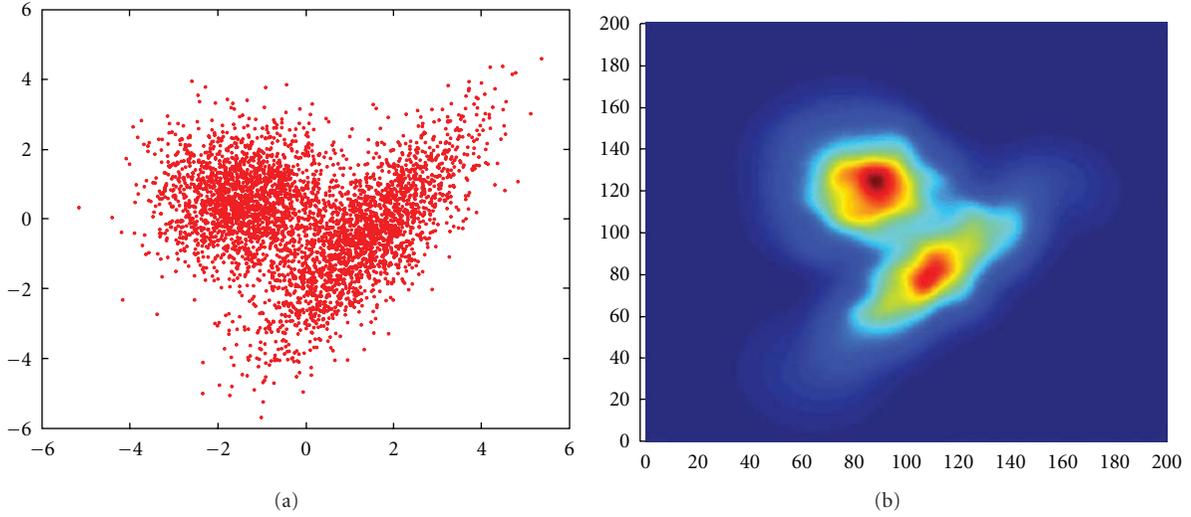


FIGURE 2: “Engytime” dataset and the related estimated density function.

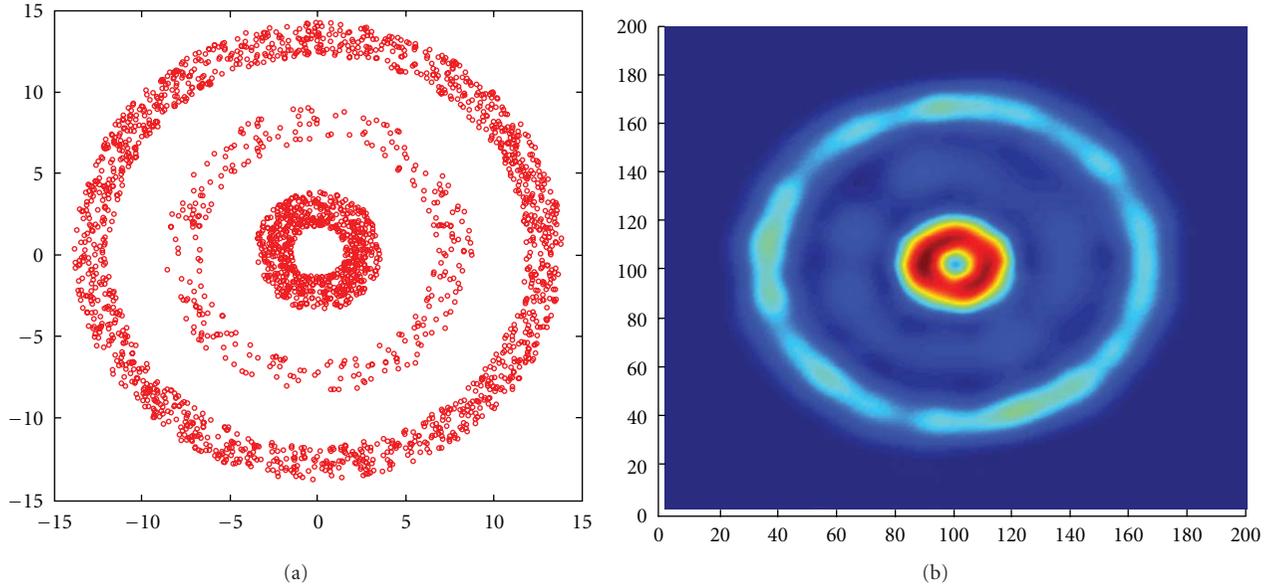


FIGURE 3: “Rings” dataset and the related estimated density function.

also represented by w_i and their neighbors. Then, h_i depends on the variability s_i computed for w_i and the distance $d_{i,j}$ between w_i and its neighbors, weighted by the number of data represented by each prototype and the connectivity value between w_i and his neighborhood.

Now, since the density D for each prototype w is known ($f(w_i) = D_i$), a gradient-descent method can be used to determine the weights α_i . The α_i are initialized with the values of D_i , then these values are reduced gradually to better fit $D = \sum_{i=1}^M \alpha_i K_i(w)$. To do this, the following criterion is optimized:

$$\alpha = \arg \min_{\alpha} \frac{1}{M} \sum_{i=1}^M \left[\sum_{j=1}^M (\alpha_j K_j(w_i)) - D_i \right]^2. \quad (5)$$

Thus, we have a density function that is a model of the dataset represented by the enriched SOM. Some examples of estimated density are shown on Figures 2 and 3.

A measure of dissimilarity between two clusters A and B can now be defined, represented by two models: $M_A = [\{w_i^A\}_{i=1}^{N^A}, f^A]$ and $M_B = [\{w_j^B\}_{j=1}^{N^B}, f^B]$, with N^A and N^B the number of prototypes representing A and B and f^A and f^B the density function of A and B .

The dissimilarity between A and B is given by

$$CBd(A, B) = \frac{\sum_{i=1}^{N^A} f^A(w_i^A) \log(f^A(w_i^A)/f^B(w_i^A))}{N^A} + \frac{\sum_{j=1}^{N^B} f^B(w_j^B) \log(f^B(w_j^B)/f^A(w_j^B))}{N^B}. \quad (6)$$

The idea is to compare the density functions f^A and f^B for each prototype w of A and B . If the distributions are identical, these two values must be very close. This measure is an adaptation of the weighted Monte Carlo approximation of the symmetrical Kullback-Leibler measure (see [17]), using the prototypes of an SOM as a sample of the database.

3. Results and Analysis

3.1. Validity of the Clustering. The effectiveness of the proposed two-level clustering method have been demonstrated in [6] by testing the performances on 10 databases presenting various clustering difficulties. DS2L-SOM was compared to S2L-SOM (similar to DS2L-SOM but using only neighborhood information) and to some traditional two-level methods, in term of clustering quality (Jaccard and Rand indexes [18]) and stability (subsampling-based method [19]). The selected traditional algorithms for comparison are K -means and Ascendant Hierarchical Clustering (AHC) applied (i) to the data and (ii) to the prototypes of the trained SOM. The Davies and Bouldin [20] index was used to determine the best cutting of the dendrogram (AHC) or the optimal number K of centroids for K -means. S2L-SOM and DS2L-SOM determine the number of clusters automatically and do not need to use this index. In AHC, the proximity of two clusters was defined as the minimum of the distance between any two objects in the two different clusters. The results for the external indexes show that for all the databases DS2L-SOM is able to find without any error the expected data segmentation and the right number of clusters. This is not the case of the other algorithms, when the groups have an arbitrary form, when there is no structure (i.e., only one cluster) in the data or when clusters are in contact. Considering the stability, the DS2L-SOM algorithm shows excellent results, whatever the dimension of data or the clusters' shape. It is worth noticing that in some cases the clustering obtained by the traditional methods can be extremely unstable.

To summarize, DS2L-SOM presents some interesting qualities in comparison to other clustering algorithms

- (i) the number of cluster is automatically detected by the algorithm,
- (ii) no linearly separable clusters and nonhyperspherical clusters can be detected, and
- (iii) the algorithm can deal with noise (i.e., touching clusters) by using density estimation.

3.2. Validity of the Comparisons. In order to demonstrate the performance of the proposed dissimilarity measure, nine artificial datasets generators and two real datasets where used in [7].

The main idea to test the quality of the comparison measure is that a low dissimilarity value is only consistent with a similar distribution and does, of course, give an indication of the similarity between the two sample distributions. On the other hand, a very high dissimilarity does show, to the given level of significance, that the distributions are different.

TABLE 1: Value of the Dunn index obtained from various dissimilarity measures to compare various data distributions.

Distributions	Average	Minimum	Ward	Proposed
Ring 1-3 + Spiral 1-2	0.4	0.9	0.5	1.6
Noise 1-4	1.1	1.4	22.0	115.3
Shuffle 1-2	1.1	16.5	6.3	27.6

Then, if the measure of dissimilarity is efficient, it should be possible to compare different datasets (with the same attributes) to detect the presence of similar distributions, that is the dissimilarity of datasets generated from the same distribution law must be much smaller than the dissimilarity of datasets generated from very different distribution. This is measured using a generalized index of Dunn [21]: the higher is this index, the better is the comparison measure (see [7]).

The results was compared with some distance-based measures usually used to compare two sets of data (here, we compare two sets of prototypes from the SOMs). These measures are the average distance (Ad: the average distance between all pair of prototypes in the two SOMs), the minimum distance (Md: the smallest Euclidean distance between prototypes in the two SOMs), and the Ward distance (Wd: The distance between the two centroids, with some weight depending on the number of prototypes in the two SOMs) [22].

As shown in Table 1, the new dissimilarity measure using density is much more effective than measures that only use the distances between prototypes. Indeed, the Dunn index for density-based measure is much higher than distance-based ones for the three kinds of datasets tested: nonconvex data, noisy data, and real data. This means that the new dissimilarity measure is much more effective for the detection of similar datasets.

3.2.1. Algorithm Complexity. The complexity of the algorithm is scaled as $O(T \times M)$, with T the number of steps and M the number of prototypes in the SOM. It is recommended to set at least $T > 10 \times M$ for a good convergence of the SOM [8]. In this study, we use $T = \max(N, 50 \times M)$ as in [10] (N is the number of data). This means that if $N > 50 \times M$ (large database), the complexity of the algorithm is $O(N \times M)$, that is, is linear in N for a fixed size of the SOM. Then, the whole process is very fast and is suited for the treatment of large databases. Also, very large databases can be handled by fixing $T < N$ (this is similar as working on a random subsample of the database).

This is much faster than traditional density estimator algorithms as the Kernel estimator [9] (that also needs to keep all data in memory) or the Gaussian mixture model [23] estimated with the EM algorithm (as the convergence speed can become extraordinarily slow [24]).

4. Case Study

In this section, an adaptation of the algorithm is introduced to deal with RFID data, and two applications are described. The first application aims at mining customers trajectories

in a supermarket during their shopping. The second is an analyze of migration behavior in an ant's colony.

4.1. Proposed Method. The proposed method is to analyze RFID signal proceeds in three steps.

- (1) Data postprocessing: The aim is to analyze the variation of the spatial behavior over time. To do that, a current spatiotemporal behavior must be defined for each instant during the following. This behavior must be inferred from the complex and noisy RFID signal. To do this, we compute for each x seconds of the tag's trajectory, a vector representing some relevant spatiotemporal information during a y minutes time window centered on the current time ($x \ll y$).

Obviously, this definition implies some correlations between the description of two time windows if they are separated by less than y minutes, as they overlap. This will allow us to detect when a customer moves from one area to another, by detecting sudden change in the description of current location.

- (2) Detection of individual homogeneous patterns: In order to regroup similar behaviors and to detect changes over time, an enriched SOM is applied on time windows from each individual sequences. DS2L-SOM is then applied on the enriched SOM as in Section 2.3, then it uses the density information to detect sudden change in the location windows, so as to regroup similar windows. The goal is to detect time-stable patterns inside the RFID signal. The signal is thus segmented in a succession of homogeneous patterns.
- (3) Detection of similar patterns: The method used in step (3) allows to analyze efficiently the spatiotemporal behavior of each tag. Now, a method is needed to compare all these individual sequences so as to perform an analysis at the collective level.

The idea is to define a similarity measure between two set of prototypes (from the enriched SOM) that represent two individual subsequences (i.e., clusters). For this job, the related density function is computed for each stable pattern as in Section 2.3. Then, a similarity matrix between each pattern of each RFID signal is computed. Finally, DS2L-SOM is applied on this matrix in order to find similar pattern in different RFID signal.

4.2. Mining Customers Trajectories. In this application, we aim at studying the individuals' spatiotemporal activity during their shopping in a supermarket. Until now, little research has been undertaken in this way. Usual questions are: how do customers really travel through the store, do they go through every area or do they skip from one area to another in a more direct manner, do they follow a single, dominant pattern, or are they rather heterogeneous?

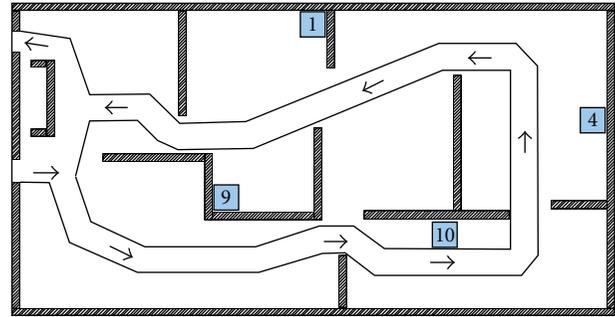


FIGURE 4: The RFID experimental device. Yellow box represents readers positions.

Tag ID	Time of the recording
0c00011629 (FF)	172.30.149.10 ; 19/02/2009 ; 10:02:52
IP address of the RFID reader	

FIGURE 5: Example of a recorded scan in the data file.

4.2.1. Context. The purpose of this work is to explore data recorded via an RFID device to model and analyze the purchasing behavior of customers. In particular, we would know the time spend in each area of the store so as to detect hot spots and cold spots. We also aim at analyzing the customers trajectory patterns.

The movement of customers during their shopping was monitored using RFID device. To do this, some plastic basket are at the disposal of the customers. Each basket have an RFID tag glued on is back.

The supermarket used for this experiment is a store specializing in the sale of decorative objects (6000 m²), with an average of 2,500 visits each month. An RFID device with four readers was installed in the different area of the store to analyze the movement of customers. The first tests carried out on site confirm that the positioning of readers is very delicate and only allow testing to find the optimum location. Figure 4 shows the final position of the readers inside the shop, noting that the readers numbers (1, 4, 9 and 10) represent the last number in their IP address. The information recorded by readers are handled by an RFID electronics and then sent to a computer which creates and stores the data files.

The data files are in text format. They indicate, for each scan (about one scan per second), the ID number of the tag detected, the IP address of the reader that have detected the tag, and the date and time of the detection (Figure 5). If, during a scan, none is detected, nothing appears in the data file.

As a customer moves inside the store, they are detected successively by different readers. However, depending of the crossed area, one tag can be detected by more than one reader, approximatively at the same time. This adds much more information about the actual location of the customer, but this also make the moving sequence much

more hard to understand. Furthermore, data recorded a very noisy, because of perturbations of the RFID signal by all the metallic structures and by human bodies (see Figure 6 for an example).

4.2.2. Analyses. Here, we want to analyze the variation of the customers spatial behavior over time. To do that, a current location must be defined. The current location represent the area where the customer is in a given time. For each 10 seconds of the customer trajectory, a vector is computed representing how many times and how long each RFID reader have detected the customer's tag during a 3-minute time window centered on the current time. This will allow to detect when a customer moves from one area to another, by detecting sudden change in the description of current location.

In order to regroup similar current location and to detect changes over time, the algorithm DS2L-SOM is applied on time windows from each individual sequences.

By using the similarity measure to compare all the subsequences of all the recorded customers in the first day of recording, we found six clusters that represent six well-defined homogeneous locations. The similarity measure can now be used to label each new customer's sequence recording after the first day. This is fast enough to be made in real time during the customer's shopping.

4.2.3. Main Results. The analysis method allowed to find six well-defined homogeneous locations (named sectors). This means that we were able to define more well-localized area than the number of reader (50% more), this is a good information extraction. The sectors can be described as follows (see also Figure 7):

- (S1) detected by reader 9 only, it correspond to the entrance of the store. Baskets waiting for new customers are detected in this sector.
- (S2) detected by reader 1 only. In this sector, customers can find flowers and vases.
- (S3) mainly detected by reader 4 and 10. In this sector, customers can find wrought iron objects.
- (S4) mainly detected by reader 1, sometime by reader 9 (wood furnitures).
- (S5) mainly detected by reader 9 and 10, sometime by reader 4 or 1 (dishes and small objects).
- (S6) mainly detected by reader 1 and 4, sometime by reader 9 (Mirrors and linens).

Figure 7 shows an estimation of the location of the different sectors. The frequency of transition between one sector to another is also computed. This gives an idea of how customers move inside the store. We can see for example that customers always take the same way in the first part of the store ((S1) and (S2)), but act more freely at the bottom of the store ((S3) to (S6)). (S5) seems to be a key area as there is many transitions from and to it, and it is connected to four over the five other sectors.

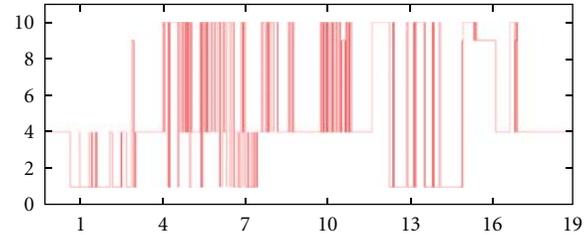


FIGURE 6: Example of a moving sequence. On the abscissa is the time (minutes). On the ordinate are the readers detecting the tag.

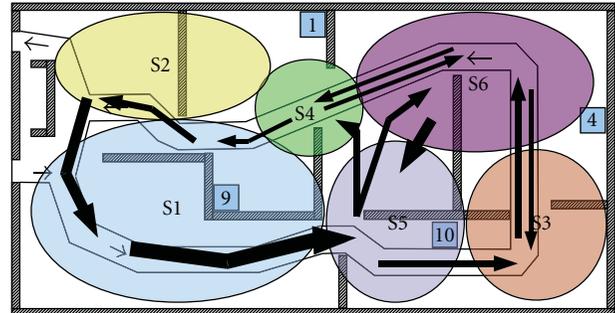


FIGURE 7: Estimation of sectors location. The thickness of arrows is proportional to transitions frequency.

Finally, the mean time spent in each sector is computed so as to find hot and cold spot. This shows us that (S5) is a very hot spot (48% of the time) unlike (S2) (6%) and (S4) (2%) which are very cold spot. Note that we do not use (S1) for this analysis, as this sector include waiting baskets.

4.3. Monitoring of an Ant Colony Migration. Animal societies are dynamic systems characterized by numerous interactions between individual members. Such dynamic structures stem from the synergy of these interactions, the individual capacities in information processing, and the diversity of individual responses.

Ants, often caricatured and little known, have nevertheless a huge ecological impact, and they are considered as major energy catalysts. Their complex underground nests contributes to soil ventilation, and because of their predatory and detritivore diets, they contribute to ecosystem equilibrium. Ant colonies face rapid changes of environmental conditions and constraints through an important individual flexibility. The following study aims at studying the mechanisms leading to a colony migrations (change of nest). Migration is a widespread phenomenon in many species, but it remains a risky event because during the movement the queen and brood will be particularly vulnerable. The strategies used in nest choice and movement organization are therefore crucial for group survival.

4.3.1. Context. An RFID device has been developed for this study. Based on marketed products, it requires little development. It consists of a network of RFID readers in a constrained space with compulsory passageways in an

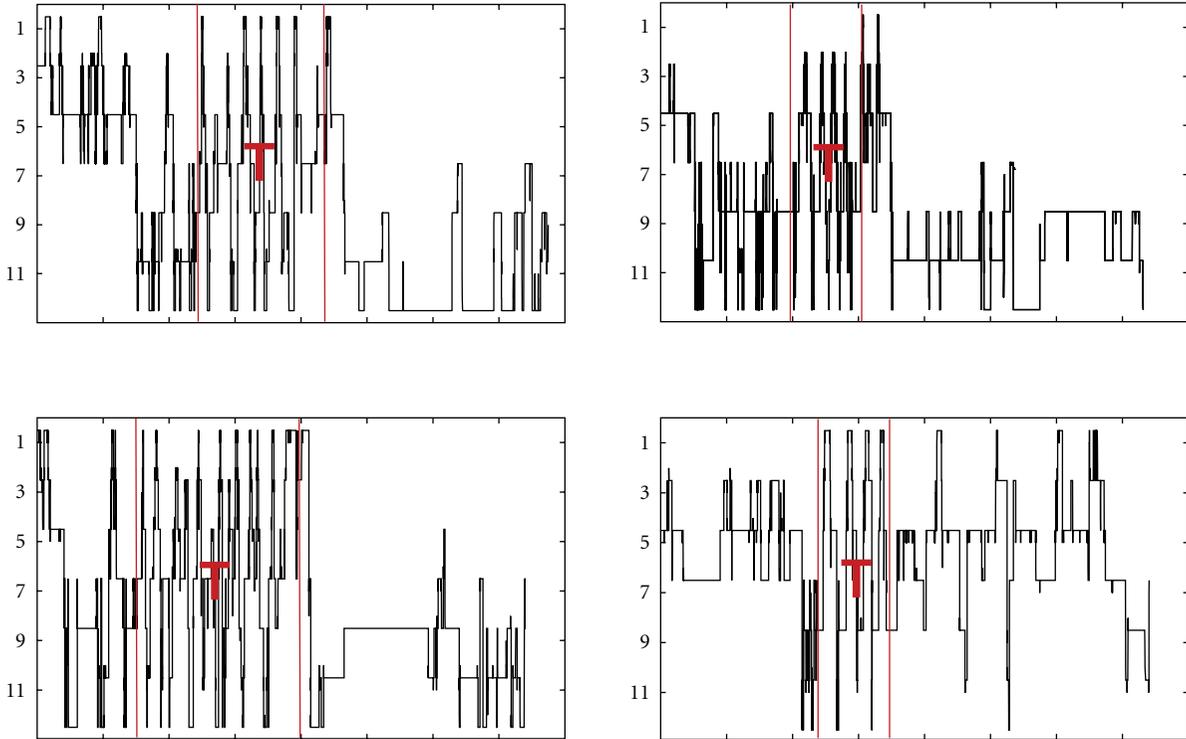


FIGURE 11: Examples of detected transportation patterns.

TABLE 2: Corresponding between visual observation and automatic analysis (number of ants).

	Transportation	Others
Activity A_5	8	2
Others	0	42

group (A_0) has been defined, which corresponds to activity segments where a clear pattern is not detectable. The activity of each ant have been labeled according to those behaviors.

In order to check the validity of the obtained results, we compared them with some visual observations from a video record. A movie camera was placed over the foraging area and every ant moving across this area was filmed. This allowed us to detect only one apparent behavior: the transportation of larva and cocoon. Each ant can be identified visually thanks to some color painted on their tag. So, we know which ant does transport and at what time this behavior occurred. We compared this with the results of the automatic analysis, and we found that all the transportations subsequences were grouped into only one activity (Activity A_5 , see Table 2 and Figure 11). Moreover, only two ants have a A_5 subsequence without having been seen transporting (i.e., less than a 5% error). This result shows that there is some reliability in the clustering found with our automatic method.

The analysis of the spatiotemporal structure of each activity have been used by ethologist for hypothesize a plausible explanation for each behavior.

- (A_0) *Unstructured patterns*. The ants are not participating into any activity and their path through the rooms is not structured.
- (A_1) *Quick exploration of the new nest*. The ants have just discovered the new nest and start a fast and active exploration of the new site.
- (A_2) *Panic movements*. This behavior is mostly expressed by nurses: due to the new environmental conditions—strong light and increase of the temperature—the old nest is no more suitable for a good brood care.
- (A_3) *Panic movements in the old nest*. It is very similar to (A_2), but the ants are more feared (this behavior seems to be expressed mainly by the youngest ants).
- (A_4) *General patrolling*. This is a general exploration of the environment.
- (A_5) *Transportation*. The ants is transporting something, that is, the queen, a cocoon, a larva, or an egg.
- (A_6) *Preparation of the new nest*. The new nest is now known to be safe, but some works are needed to prepare the nest for an optimal brood care.
- (A_7) *Patrolling in the old nest*. This behavior is a defensive patrolling inside the old nest in react to the disturbance.
- (A_8) *Patrolling in the new nest and foraging area*. This behavior is a defensive patrolling inside the new environment.

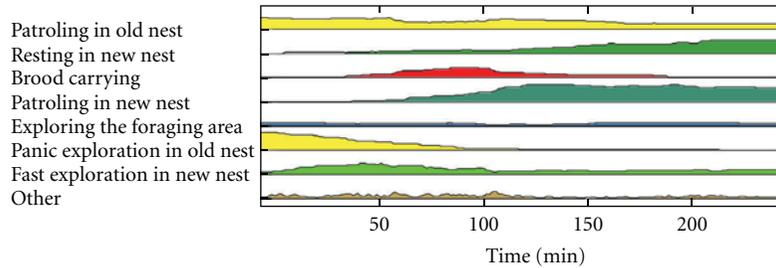


FIGURE 12: Activity analysis: evolution of the number n_{ij} of individuals involved in the activities $A_0 \dots A_2$.

Now, the abstracted individual activity during emigration can be used to compute the collective dynamic of emigration. Figure 12 is a representation of this dynamic, showing the number of ants expressing each activity during the migration process. As we can see, colony emigration follows a typical pattern: when the light is switched on, the first event is a panic exploration of the old nest that is expressed by most of the ants. Slower patrolling will remain constant during all the process and concern not only the old nest but also the foraging area. The second event is the discovery of new nests followed by the brood transport. Afterwards, a more constant exploration of the new nest occur. The last activity, which appears gradually, is the settlement in the new nest.

From an ethological point of view, these results are of great help for understanding how tasks are distributed during a nest relocation. In fact, we obtained a very accurate description of the dynamic of the whole colony during all the emigration phase allowing us to emit some strong hypothesis about the function of the different behavior during the nest relocation phase. Some results are in accordance with previous works, especially the behaviors that can be observed in the foraging area. For example, the dynamic of the transportation behavior detected by the system match up the results presented in [26]. These hypotheses should now be validated by repeating the experiment with different colonies and different species. A complete understanding of the emigration process based on systematic experimentations would be an important step ahead for the research in social insects.

5. Conclusions

In this paper, a new algorithm is proposed for modeling data structure, based on the learning of an SOM, and a measure of dissimilarity between cluster structures. The advantages of this algorithm are not only the low computational cost and the low memory requirement, but also the high accuracy achieved in fitting the distribution of the modeled datasets. The results obtained on the basis of artificial and real datasets are very encouraging. The new unsupervised algorithm used in this paper is an efficient data mining tool for behavioral studies based on RFID technology. It allows discovering and comparing stable patterns in a RFID signal and is suitable for continuous learning.

Here, it was possible to highlight some characteristics of spatial organization of customers during their shopping in a big store from complex and noisy spatiotemporal database. Moreover, the characteristics of spatiotemporal organization in ant colonies during migration were described, and these results are perfectly compatible with the results of previous works using classic methods [27].

Acknowledgment

This work was partly supported by the ANR (Agence National de la Recherche) CADI 07 TLOG 003 and SILLAGE 05 BLAN 017701.

References

- [1] P. Lyman and H. R. Varian, "How Much Information, 2003," <http://www.sims.berkeley.edu/how-much-info-2003>.
- [2] J. Gehrke, F. Korn, and D. Srivastava, "On computing correlated aggregates over continual data streams," *Special Interest Group on Management of Data Record Record*, vol. 30, no. 2, pp. 13–24, 2001.
- [3] G. S. Manku and R. Motwani, "Approximate frequency counts over data streams," in *Very Large Data Base*, pp. 346–357, 2002.
- [4] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *Proceedings of the 6th SIAM International Conference on Data Mining*, pp. 328–339, April 2006.
- [5] C. Aggarwal and P. Yu, "A survey of synopsis construction methods in data streams," in *Data Streams: Models and Algorithms*, C. Aggarwal, Ed., pp. 169–207, Springer, 2007.
- [6] G. Cabanes and Y. Bennani, "A local density-based simultaneous two-level algorithm for topographic clustering," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 1176–1182, June 2008.
- [7] G. Cabanes and Y. Bennani, "Comparing large datasets structures through unsupervised learning," in *Proceedings of the International Conference on Neural Information Processing*, pp. 546–553, 2009.
- [8] T. Kohonen, *Self-Organizing Maps*, Springer, Berlin, Germany, 2001.
- [9] B. Silverman, "Using kernel density estimates to investigate multimodality," *Journal of the Royal Statistical Society, Series B*, vol. 43, pp. 97–99, 1981.
- [10] J. Vesanto, "Neural network tool for data mining: SOM Toolbox," 2000.

- [11] S. Sain, K. Baggerly, and D. Scott, "Cross-validation of multivariate densities," *Journal of the American Statistical Association*, vol. 89, pp. 807–817, 1994.
- [12] A. Ultsch, "Clustering with SOM: U*C," in *Proceedings of the Workshop on Self-Organizing Maps*, pp. 75–82, 2005.
- [13] E. E. Korkmaz, "A two-level clustering method using linear linkage encoding," in *Proceedings of the International Conference on Parallel Problem Solving From Nature*, vol. 4193 of *Lecture Notes in Computer Science*, pp. 681–690, 2006.
- [14] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.
- [15] S.-H. Yue, P. Li, J.-D. Guo, and S.-G. Zhou, "Using Greedy algorithm: DBSCAN revisited II," *Journal of Zhejiang University Science*, vol. 5, no. 11, pp. 1405–1412, 2004.
- [16] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.
- [17] J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 4, pp. 317–320, 2007.
- [18] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Cluster validity methods," *Special Interest Group on Management of Data Record Record*, vol. 31, no. 2-3, pp. 40–45, 2002.
- [19] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," in *Proceedings of the Pacific Symposium on Biocomputing*, vol. 7, pp. 6–17, 2002.
- [20] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, 1979.
- [21] J. C. Dunn, "Well separated clusters and optimal fuzzy partitions," *Journal of Cybern*, vol. 4, pp. 95–104, 1974.
- [22] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Upper Saddle River, NJ, USA, 1988.
- [23] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.
- [24] H. Park and T. Ozeki, "Singularity and slow convergence of the em algorithm for gaussian mixtures," *Neural Processing Letters*, vol. 29, no. 1, pp. 45–59, 2009.
- [25] U. Galassi, A. Giordana, C. Julien, and L. Saitta, "Modeling temporal behavior via structured hidden markov models: an application to keystroking dynamics," in *Proceedings of the Indian International Conference on Artificial Intelligence*, pp. 2140–2154, 2007.
- [26] A. Pezon, D. Denis, P. Cerdan, J. Valenzuela, and D. Fresneau, "Queen movement during colony emigration in the facultatively polygynous ant *Pachycondyla obscuricornis*," *Naturwissenschaften*, vol. 92, no. 1, pp. 35–39, 2005.
- [27] D. Fresneau and P. Dupuy, "Behavioural study of the primitive ant *Neoponera apicalis*," *Animal Behaviour*, vol. 36, no. 5, pp. 1389–1399, 1988.