A NEW TECHNIQUE FOR SUMMARIZING VIDEO SEQUENCES THROUGH HISTOGRAM EVOLUTION

Tao Wan[1] *and* *Zengchang Qin*[2]*

[1]Department of Electrical and Electronic Engineering
University of Bristol, Bristol, BS8 1UB, UK
[2]Intelligent Computing and Machine Learning Lab
School of Automation Science and Electrical Engineering
Beihang University, Beijing, 100191, China

ABSTRACT

In this paper, we present an efficient technique based on histogram evolution for summarizing video sequences to make them more amenable to browsing and retrieval. First, a ground-truth database of videos is generated in which the shot breaks for each video are detected by human subjects and numbered in order. Three types of histogram are then used to capture the characteristics of color content containing in the video frames. The principle components analysis (PCA) method is adopted to reduce the histogram dimensions and form a 2D feature space. Finally, two approaches, frame difference measures and Fuzzy C-means clustering, are employed to extract video shot breaks. Polylines are drawn between the detected shot breaks to show that the histogram of their colors evolves from frame to frame. In comparison with the ground-truth database, the proposed algorithm achieves a surprising high detection accuracy rate. The extensive experiments also demonstrate that the patterns of histogram evolution can be useful to identify the shot break types, such as cut, dissolve, fade-out, fade-in, and wipe.

Index Terms— video summarization, shot boundary detection, histogram evolution

1. INTRODUCTION

Video summarization is an important research area for studying video applications including video indexing, browsing and retrieval. A concisely and intelligently generated video summarization will not only enable a more informative interaction between human and computer during the video browsing, but also help to build more meaningful and quicker video indexing and retrieval systems.

A video summary is a sequence of still images or short clip representing the content of a video in such a way that the target part is rapidly provided with concise information about the video while the essential message of the original is well preserved. Theoretically, a video summarization can be performed manually [1] and automatically [2, 3, 4]. Due to the huge volumes of video data and limited manpower, it is more and more crucial to develop fully automated video analysis and processing tools so as to reduce the human involvement in the video summarization process. This thereby motivates our research on automatic video summarization.

There is much literature focusing on automatic video summarization for a particular type of video, especially for sport programmes. In [4], Ekin *et al*. presented a fully automatic framework for analysis and summarization of soccer video using cinematic and object-based features. Their method is robust over a large data set captured in different countries and under different conditions. In [5], a statistical method is described to explore the specific spatial and temporal structure of highlights in baseball game video. Moreover, in [6], the researchers derived a unified approach for video summarization in which scene modeling is achieved by normalized cut algorithm and temporal graph analysis. A temporal graph can inherently represent the evolution and perceptual importance of a video. Fu and Zeng [7] posed a shot boundary detection method based on the histogram difference of local images. It is able to detect both abrupt transitions and gradual transitions. Compared to previous work, our proposed algorithm enables to discover various types of shot break using histogram evolution between the video frames. By examining the polyline through the detected key frames, we learn the patterns associated to the different types of shot break, such as cut, dissolve, and wipe etc., thus leading to an accurate way for summarizing videos. It is worth noting that our proposed framework can be applied to a variety of video types which have their own characteristics.

The remainder of the paper is organized as follows: Section 2 provides the details to generate a ground-truth database containing different types of shot break through human judgment. In Section 3, a histogram computation and dimension reduction problem is described. A shot break detection process and simulation results are presented in Section 4 and Section 5, respectively. Finally, conclusions and suggestions for future work are given in Section 6.

2. GROUND TRUTH GENERATION

The purpose for establishing a ground-truth database is in two aspects. One is to evaluate the performance our proposed video summary algorithm comparing with human perception. The other is to establish a basis for learning the polyline patterns relating with various shot break types. In general, it is very difficult telling the shot changes during normal playback time by the human eyes, even slow down the playback speed. In order to eliminate this effect to the final detection results, we extract overall frames for each experimental video and store them in a specified directory. Each examiner was asked to inspect these frames one by one in a consecutive order. There are twelve video sequences with different lengths used to

* Corresponding author, email address: zcqin@buaa.edu.cn, is partly funded by NCET

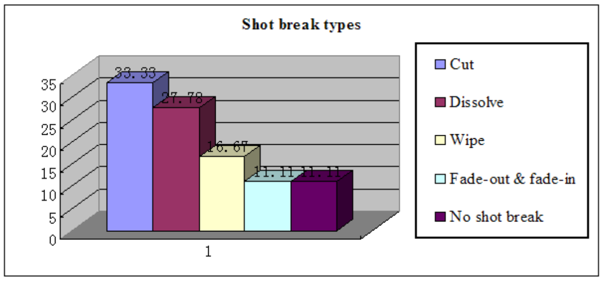


Fig. 1. Percentage of shot break types in the total test video clips.

construct the database. The video descriptions are listed in Table 1.

Table 1. Descriptions of the 12 original video sequences

Index	Sequence Name	Image Type	Length (frame)	Content
1	ad.avi	true-color	644	advertisement
2	movie.avi	true-color	3363	trailer
3	people.avi	true-color	192	concert
4	sta.avi	true-color	317	concert
5	walk.avi	grayscale	300	outdoor wildlife
6	run.avi	grayscale	74	news clip
7	logo.avi	grayscale	400	advertisement
8	car.avi	grayscale	184	documentary
9	dissolve.avi	true-color	270	sea life
10	md.avi	true-color	185	trailer
11	tailer.avi	true-color	300	trailer
12	news.avi	true-color	403	news clip

The frame-by-frame inspection generates raw sequences from which our test video segments are created. They are new sequences that are spliced together by a random choice from individual ground-truth video. Each splice point is also randomly chosen from these twelve video sequences. The extracted clip length is automatically computed but subject to the constraint that the sequence does not leap over any scene cuts. This means the selected frames are captured under the same scene. Fig. 1 illustrates a bar chart showing the percentage of the shot break types in the total test video clips. Cut occupies the highest percentage since it is a most common shot break type appearing in the natural video sequences.

3. HISTOGRAM COMPUTATION AND DIMENSION REDUCTION

Histogram has been widely used in the field of video summarization to describe the color features of multiple video frames [8, 7]. We adopt three types of histogram and explore their impact on the final 2D feature space construction. The color histogram provides distribution information of colors in each video frame. It is computed over the RGB color channels. The grayscale histogram is applied to the graylevel images showing the the intensity changes rather than RGB colors. The average histogram calculates the average value of RGB to maintain the color distribution while decrease amount of data to be processed. A PCA method [9] is implemented to reduce the dimension of the histogram features. Therefore, each frame would be a single point to be plotted in a 2D feature space.

Here, we consider a true-color frame (624 × 480 $pixels$) represented by a 10-bar RGB histogram. A one-dimensional vector is used to store the frequency of the histogram which has dimensions of $D = 480 \times 10 \times 3 = 14400$. Let $x_1, x_2, ..., x_{200}$ be the 200 sample video frames. For each frame:

$$Z_i = x_i - m \qquad i = 1...200 \tag{1}$$

where m is the mean image computed from these 200 images. A is defined as a matrix whose columns are the mean-subtracted sample images. $A = [Z_1, Z_2, ..., Z_{200}]$. We know that A is a 14400 × 200 matrix and its covariance matrix ($Cov(A) = 1/nAA^T$) is as big as 14400 × 14400. It is too large to compute its eigenvectors numerically. Through a few straightforward transforms, the problem could be simplified as:

$$\begin{aligned} (A^T A)v_i &= l_i v_i \\ A(A^T A v_i) &= A(l_i v_i) \\ AA^T(Av_i) &= l_i(Av_i) \end{aligned} \tag{2}$$

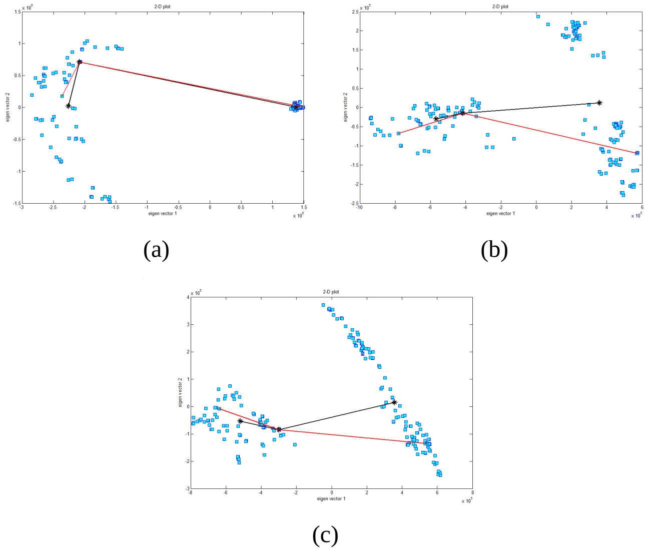
where v_i is an eigenvector of $A^T A$ and l_i is the corresponding eigenvalue. Thus, from Equation 2 the eigenvectors of $Cov(A)$ can be simply computed as $Av_i, Av_2, ..., Av_{200}$. This greatly improves the computational efficiency.

After histogram computation and PCA dimension reduction, only two main eigenvectors with the largest eigenvalues are saved to represent each frame. In order to understand how these three types of histogram affect the feature space construction, they are applied to a test video clip containing only one shot break. The first part of the sequence is taken with still background and the second part has more changeable scene. The RGB histogram is adopted for all the true-color video sequences. The grayscale histogram is only used when RGB values are not available.

4. DETECTION PROCESS

In this section, we aim to discover the shot breaks within the test video sequences through two approaches. One is a conventional method to measure the dissimilarity between two frames. The other is a Fuzzy C-means clustering method [10] which is a classical and popular algorithm for classification as well as for pattern recognition and image processing.

The general idea about frame difference measure is to compute the histogram dissimilarity from one frame to the consecutive one. As mentioned before, histogram features for each frame have been reduced to two main values and they are plotted into a 2D coordinate space. In the sense, the relative locations of the plotted points indicate the difference between the frames. It is reasonable to compute the Euclidean distance for each pair of points to measure the similarity of the corresponding frames as well as the scene changes. This method is easy to implement on the basis of 2D space. The key issue is to assign a proper thresholding in the detection process.



(a)

(b)

(c)

Fig. 2. Polyline patterns for cut. (a) Color histogram. (b) Grayscale histogram. (c) Average histogram.

At the beginning we have no prior knowledge about the video content except the plotted coordinate values. Manually adjusting the thresholding is also a time-consuming task. Here we designed an automatic thresholding value generation approach. The threshold is initialized by averaging the maximum and minimum values of the distance. The value is increased or decreased into half according to a certain rule till a short break is detected.

The Fuzzy C-means clustering method considers the scene boundary detection task as a classification problem. The analogous frames are grouped together as one cluster with a membership value assigned to each frame. The values 0 and 1 denote no membership and full membership, respectively. In general, each frame has a grade of membership between 0 and 1 indicating partial relation in the cluster. It is well known that the cluster validity problem is difficult. We solved this issue by sampling the video sequences. First, we computed an objective function using 12 short video clips randomly generated from the 12 original video sequences which contain only one shot break. We then obtained the objective function values for each video sequence. Finally, the test video clips with one or multiple shot breaks are processed by the Fuzzy C-means method iteratively till the objective function value is greater than a predefined threshold.

Shot breaks can be discovered by using the clustering results. Each cluster represents a number of frames with similar scene attributes. Once a following frame moves to the other cluster this means a scene change may happen. The edge point in the cluster can be found as a point located furthest from the cluster center. The grade membership obtained from the Fuzzy C-means clustering is used to measure the dissimilarity. The frame with the smallest membership value is our detected shot boundary. Although the clustering information is useful for finding shot breaks, it is not always feasible since some video clips may have some clusters overlapping each other. If this happens, the polylines connecting each cluster center are employed to resolve the matter. These are discussed in the following section.

(a)

(b)

Fig. 3. A summary of the video for cut. (a) Frame 450 (the frame before the short break). (b) Frame 525. (c) Frame 525 (the frame right after the short break).

5. SIMULATION RESULTS AND DISCUSSIONS

The experiments employe five types of shot break as shown in Fig. 1. Cut is an ordinary type demonstrating fast scene change. Dissolve is usually adopted in the movies and documentaries which has transition period changing from one scene to another scene. These two scenes overlap each other until previous scene disappears and the next scene becomes clear. Wipe is similar to dissolve. The difference is that two scenes do not overlap each other in the wipe. Fade-out and fade-in generally appear together. The previous scene disappears slowly and turns into black scene. This procedure is called fade-out. The scene then changes from black to the next scene. This is fade-in. Each test video clip may contain no shot breaks or have more than one type of shot break. We intend to analyze the patterns of the video frame evolution through the polylines drawn from the shot breaks and build the connection to the different types of shot break. In the following part, we display four examples using four typical video clips to evaluate the performance of the proposed algorithm.

5.1. Cut Experiment

The experimental video has totally 195 frames and contains one cut according to the ground-truth database. Fig. 2 illustrates the polyline patterns formed from the three types of histogram. The line without star is computed using the frame difference measure. The end points of the polyline are the first and last frame of the video. The middle one shows the detected scene boundary. The line with star is calculated by the Fuzzy C-means clustering. The two extremes are cluster centers and the middle star is the scene boundary. We connect each key point to display the tendency of the scene change. By inspecting the figure, we conclude that the video has two scenes and a quick change between these two scenes. In this case, the frame difference measure and Fuzzy C-means clustering achieve the same shot break location. A summary of the video content is generated (see Fig. 3) utilizing the extracted shot break frames.

5.2. Dissolve Experiment

In this example, a more complicated video clip is exploited that contains four dissolves and one background change within 192 frames. The plotting image with polylines is shown in Fig. 4. There are 6 clusters found by the Fuzzy C-means clustering method and Frame 30, 144, 155, 170, 171, 172 are detected as shot breaks. From the frame difference measure, we obtained slightly different results (difference within 2 frames) that Frame 28, 29, 30, 146, 151, 155, 171, 172, 173 are the shot breaks. By examining the shape of the polyline

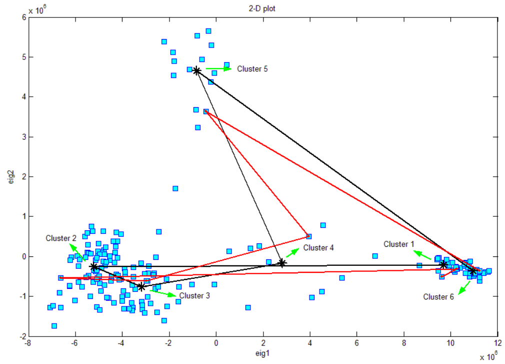
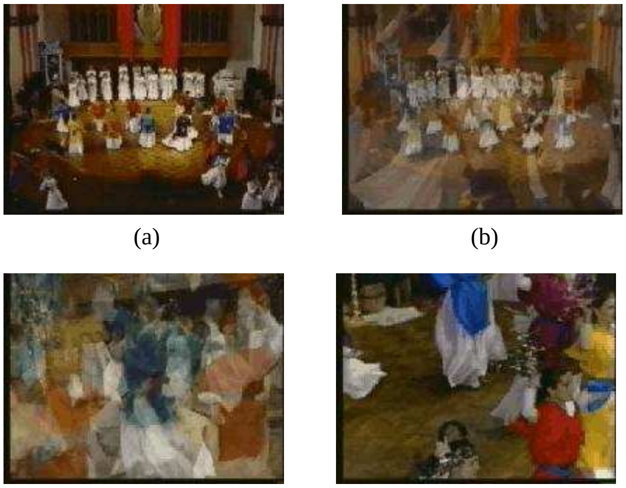


Fig. 4. Polyline pattern using color histogram.


(a)

(b)

(c)

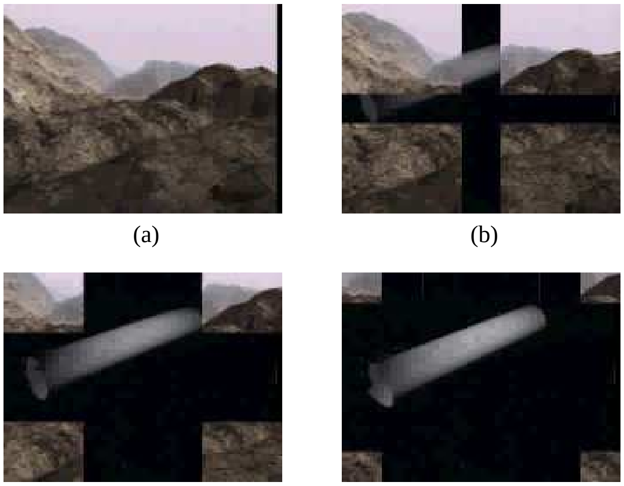
(d)

Fig. 5. A summary of the video for dissolve. (a) Frame 1. (b) Frame 30. (c) Frame 144. (d) Frame 155.

only, we know that the video sequence has five shot breaks. Furthermore, the first and the last cluster are close to each other in position which indicates that they may contain the similar scene. Combining these two results, we create a short summary for the test video as Fig. 5 shows. Dissolve is more difficult to process in comparison with cut since it has transition frames during the scene change period and each frame has similar background and objects.

5.3. Wipe Experiment

As stated before, the wipe is similar to the dissolve since there is a transition period between two scenes. This can be seen in Fig. 6 in which Frame 32 to Frame 39 represent the transition course within the video. They are plotted into a 2D feature space shown in Fig. 7 (a). The video sequence contains only one wipe so that only two classes are found by the Fuzzy C-means clustering method. Fig. 7(b) demonstrates the evolution of the plotted points from Cluster 1 to Cluster 2. By analyzing the patterns of the polylines only, we cannot tell the wipe type from the dissolve type. For this case, we extracted Frame 34 (a transition frame) to check the shot break type.

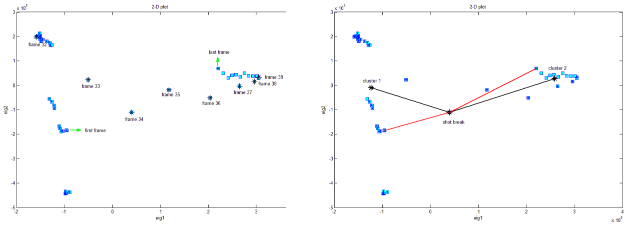

(a)

(b)

(c)

(d)

Fig. 6. An example of wipe. (a) Frame 32. (b) Frame 33. (c) Frame 35. (d) Frame 37.



(a)

(b)

Fig. 7. 2D plotting. (a) wipe frames. (b) polyline pattern.

5.4. Fade-out and Fade-in Experiment

Fade-out and fade-in is an important type of shot break often appearing in the movies. We show an example extracted from a movie trailer that consists of 70 true-color frames. Fig. 8 displays the plotting image as well as the polylines drawn by the frame difference measure and Fuzzy C-means clustering methods. The polylines demonstrate the move trend of the frames when the video is played. Moreover, the pattern of the polyline is quite different from the previous ones by cut, dissolve or wipe. A few key frames are extracted to form a short video summary in Fig. 9.

Taken it as whole, the feature space thresholding and clustering number are the two key parameters in the proposed algorithm. The thresholding value depends on the visual content of the video. It can be automatically generated through self-adjusted mechanism. Clustering number assignment is essential for performing the Fuzzy C-means clustering method. In the method, this parameter is initialized and computed based on the grade membership of the plotted frames in the 2D feature space. Fig. 10 demonstrates a bar chart of the detection accuracy evaluation indicating that the proposed algorithm achieves 86.39% accuracy rate over 50 test video sequences.

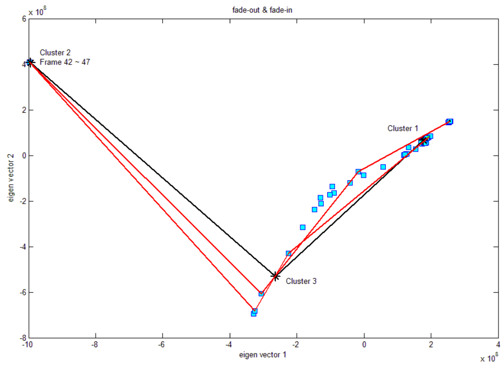
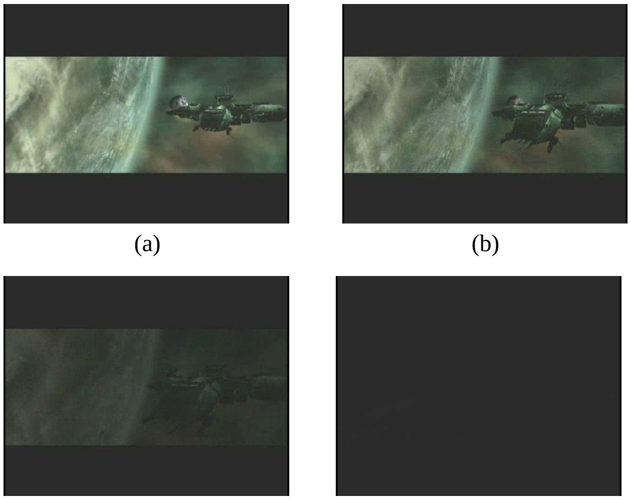


Fig. 8. Polyline pattern using color histogram.



(a) (b)

(c) (d)

Fig. 9. A summary of the video for fade-out and fade-in. (a) Frame 1. (b) Frame 25. (c) Frame 34. (d) Frame 47. (e) Frame 48. (f) Frame 70.

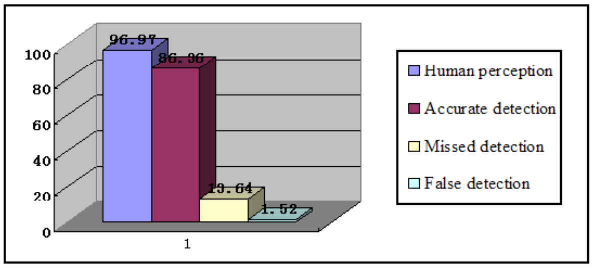


Fig. 10. Shot break detection results.

6. CONCLUSIONS AND FUTURE WORK

We have presented a new video summarization technique based on the histogram evolution. In this study, we focus on two main subjects. One is to detect occurrence of shot breaks within a video clip and create a short summary to describe the visual content of the video. The other is to analyze the polylines drawn from frame to frame and learn the patterns of the histogram evolution for different types of shot break. Two detection methods, frame difference measure and Fuzzy C-means clustering, have been employed in the proposed algorithm. The former is a validate method when notable scene changes happen. However, when the variation between the frames are small due to slow motion of the objects in the video, false and missed detection is more likely to occur. The later works better in this circumstance but fails when the neighboring clusters overlap. We aim to develop a simple and efficient video summarization method in which the jointed polylines between the detected shot breaks provide an effective way to understand and interpret the underlying content changes of a playing video clip. The experimental results demonstrate that the tendency of frame evolution can be represented by the histogram features plotted into a 2D feature space with joint polylines.

It is well known that content-based video retrieval is a challenging problem and has been the subject of significant research in the recent past. By integrating our posed algorithm with improved feature extraction and classification techniques, we are able to build an effective retrieval system to facilitate the searching of video sequences.

7. REFERENCES

[1] A. K. Elmagarmid, H. Jiang, A. A. Helal, A. Joshi, and M. Ahmed, *Video database system: Issues, Products, and Applications*, Kluwer Academic Publishers, Boston, 1997.

[2] F. Shipman, A. Girgensohn, and L. Wilcox, "Creating navigable multi-level video summaries," in *IEEE Internation Conference on Multimedia and Expo*, 2003, pp. 753–756.

[3] S. Benini, A. Bianchetti, R. Leonardi, and P. Migliorati, "Extraction of significant video summaries by dendrogram analysis," in *IEEE Internation Conference on Image Processing*, 2006, pp. 133–136.

[4] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Tran. Image process.*, vol. 12, no. 7, pp. 796–807, Jul. 2003.

[5] P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden Markov models," in *International Conference on Image Processing*, 2002, pp. 609–612.

[6] C. W. Ngo, Y. F. Ma, and H. J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Tran. Circuits and Syst. for Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.

[7] X. Fu and J. X. Zeng, "An effective video shot boundary detection method based on the local color features of interest points," in *International Symposinm of Electronic Commerce and Security*, 2009, pp. 25–28.

[8] N. Vasconcelos and A. Lippman, "Bayesian video shot segmentation," in *Neural Information Processing Systems*, 2000, pp. 1009–1015.

[9] I. T. Jolliffe, *Principal Component Analysis*, Springer, New York, 2002.

[10] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-fuzzy And Soft Computing*, Prentice Hall, Upper Saddle River, NJ, 1997.