Visual Abstraction of Wildlife Footage using Gaussian Mixture Models

David Gibson Neill Campbell Barry Thomas
Department of Computer Science
University of Bristol
Bristol
BS8 1UB, United Kingdom
{gibson,campbell,barry}@cs.bris.ac.uk

Abstract

In this paper, we present a novel approach for clip-based key frame extraction. Our framework allows both clips with subtle changes as well as clips containing rapid shot changes, fades and dissolves to be well approximated. We show that creating key frame video abstractions can be achieved by transforming each frame of a video sequence into an eigenspace and then clustering this space using Gaussian Mixture Models (GMMs). An iterative process computes a GMM configuration that best clusters the data based on a maximum likelihood threshold. The image nearest to the centres of each of the GMM components are selected as key frames. Unlike previous work this technique relies on global video clip properties and results show that the key frames extracted give a very good representation of the overall clip content. We show that, by using a single threshold, an operator can easily control the number of representative key frames generated. We also demonstrate that clustering in eigen-time space improves the video abstractions in a quantifiable manner and we demonstrate the application of this technique on a database of 307 *clips of wildlife footage containing dissolves, shot changes, fades, pans, zooms and a wide range of animal behaviours.*

1. Introduction

As computational power and storage capacities increase the creation and use of video databases becomes more extensive. The BBC's Natural History Unit in Bristol has a library consisting of hundreds of thousands of hours of wildlife footage on digital beta tape, the world's largest archive of this kind. To use (or reuse) this material currently involves a costly manual process initially requiring an archivist to enter text descriptions and timecodes of the footage into a database. When accessing the footage the desired type of content is searched via a text database which generally gives a number of candidate tapes. These are then sent (often posted around the world) to the program researchers who traverse and view the tapes looking for appropriate content. This process is repeated many times until the desired content, 'look' and 'feel' of the footage is found. The motivation for this work is in the context of investigating techniques to automatically facilitate archiving, retrieval and searching of a digitised version of this wildlife footage.

The initial database created for our work consists of over 100,000 (around 70 minutes) 30-bit YUV uncompressed PAL(601) frames. These constitute 307 clips, generally between 100 and 1000 frames which often contain more than one shot. The clips contain no audio since this is almost always added in post-production for wildlife films. The clips have few post-production editing effects and content ranging from original cellophane film reeds put on tape via a telecine machine to first cut programme and film segments. Given the nature of this database, this paper focuses on the extraction of a small number of key frames from each clip such that these key frames give a useful and meaningful representation of the overall clip content for use by an editor or producer.

1.1. Previous Work

Previous and related work regarding video analysis includes shot detection, content-based video abstraction and video skimming. Many applications require that video is divided up into component shots so as to facilitate further processing such as indexing, key frame extraction and shot content analysis. Techniques including colour histogram, motion and frequency analysis are used to determine when shot changes have occurred, representative key frames are then extracted from individual shots [6, 7, 5]. Video skimming introduces the use of audio and language analysis to generate condensed representations of the original material [8]. In our approach we are not concerned with shot segmentation, just that the visual abstraction generates a small number of informative key frames. This then facilitates fast and useful browsing of a large database of wildlife video footage.

In [9] image histograms and inter-frame motion features have been used as the basis of an eigenspace in order to ob-

tain a high level video 'story' structure. Due to the nature of our database, we have found these techniques are not suitable for key frame abstraction, as demonstrated by Figure 1. In the top-left image, a giraffe walks towards camera, partially obscured by a heat-haze. In the top-right frame, a bird in flight suffers from motion blur. Both of these sequences have proved unsuitable for motion tracking. In the bottom images, an orang-utan hides in a canopy of leaves. The colour histogram of this sequence changes little throughout the shot despite containing both animal and camera motions. Many of our sequences include tracking shots of animals where, once again, little or no change in colour content is observed. In this work we cluster the eigenspace of all images in a clip using a Gaussian Mixture Model.

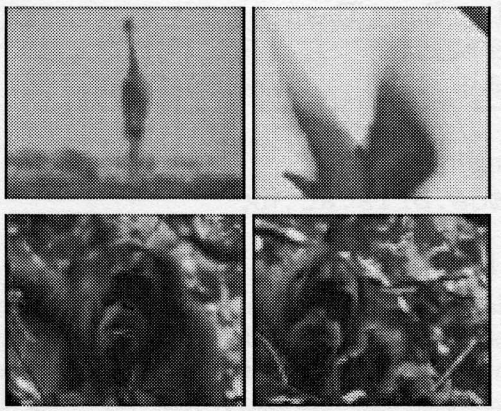

Figure 1: Some images from our wildlife video database. Traditional techniques, such as motion analysis, fail due to heat haze (top left), motion blur (top right). Other techniques, such as histogramming, fail when motions occur with little or no change in colour content (bottom).

Principal Components Analysis, PCA, (also known as eigen analysis) has been used extensively in computer vision for image reconstruction, pattern matching [2] and classification [3]. The application of PCA here is to perform dimensionality reduction so that the high-dimensional image space can be represented by a much lower dimensioned space whilst retaining the significant variations of the original. By using images as input to PCA, a whole clip can be represented by a set of low dimensional points that encode the most significant changes in illumination across the clip [1]. These changes are caused by various events such as camera motions, object motions, shot changes and so on. In this work, after clustering is performed, key frames are then identified.

In [10] eigen analysis was used to determine shot changes in video. Each frame was represented as a point in a low-dimensional space. Significant shot changes result in rapid moves across this space. Thresholds were set on the direction and magnitude of these jumps and shot boundaries obtained when such changes are significant. Key frames were computed from the mean image of each shot. This approach works well for some sequences in our database, such as that shown in Figure 2. Here the clip is comprised of three distinct shots, forming three well defined clusters in the two-dimensional eigenspace. However, other sequences would not be suitable for such an approach. For instance, in Figure 3, no large moves across the eigenspace occur during a sequence in which a camera zooms in on a tree. The trajectory through eigenspace is subtle, smoothly varying and devoid of any rapid changes.

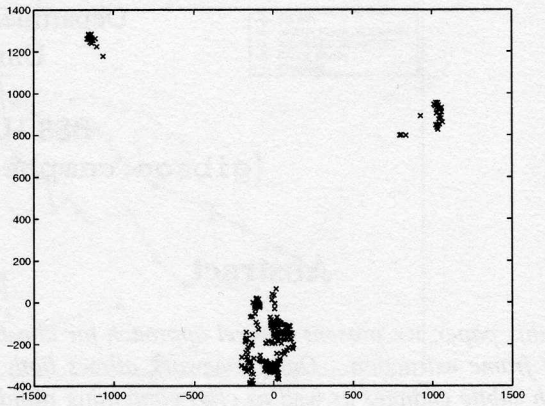

Figure 2: A plot of the responses to the first two principal modes (X-axis : Mode 1, Y-axis : Mode 2) for each frame of a wildlife sequence consisting of three major shot changes.

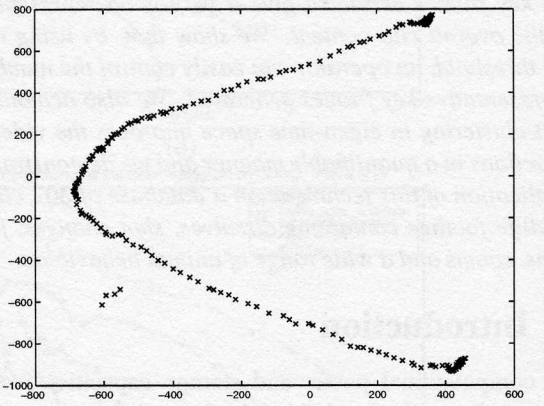

Figure 3: A plot of the responses to the first two principal modes for each frame of a sequence in which a camera zooms in on a tree. The clip consists of 262 frames.

For this reason we propose a method that will handle both of the above examples within a consistent framework. We cluster the eigenspace using a fully covariant Gaussian

Mixture Model (GMM) [4] and use the frame corresponding to the centre of each component as a key frame.

The rest of this paper is organised as follows. Section 2 details the techniques used. Section 3 gives results and discusses issues related to the implementation and application of this technique. Section 4 gives a summary and our conclusions.

2. Methodology

Given the i^{th} image in a sequence of images, each of which consists of ρ pixels, we form the vector x_i by concatenating the pixels of the image in raster scan order and removing the mean image of the sequence. The matrix X is created using the x_i's as column vectors. Traditionally, the principal modes, q_i, are extracted by computing

$$XX^T q_i = \lambda_i q_i \tag{1}$$

where λ_i's are the eigenvalues, a measure of the amount of variance each of the eigenvectors accounts for. Unfortunately, the matrix XX^T is typically too large to manipulate since it is of size ρ by ρ. Such computation is wasteful anyway since only N principal modes are meaningful, where N is the number of example images. In this work $N << \rho$. Therefore we compute:

$$X^T X u_i = \lambda_i u_i \tag{2}$$

and we can obtain the q_i's that we actually require using:

$$q_i = X u_i \tag{3}$$

In practice only the first d modes are used, $d < 10 << N$.

The frame responses to the eigenspace are modelled as a mixture of M, d-dimensional Gaussians in the form:

$$p(x) = \sum_{j=1}^{M} p(x|j)P(j) \tag{4}$$

where $P(j)$ corresponds to the prior probabilities that a given response, x, was generated by the j^{th} component and:

$$p(x|j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} e^{\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\}} \tag{5}$$

where μ_j and Σ_j are the mean and covariance matrix for each component. Posterior probabilities are given by Bayes' theorem in the form:

$$P(j|x) = \frac{p(x|j)P(j)}{p(x)} \tag{6}$$

The above equations can then be used to derive the update functions 7, 8 and 9, which are used in the Expectation Maximisation algorithm to train the model. To update the mixture centres we use:

$$\mu_j^{new} = \frac{\sum_n P^{old}(j|x^n)x^n}{\sum_n P^{old}(j|x^n)} \tag{7}$$

To update the mixture covariances:

$$\Sigma_j^{new} = \frac{1}{d}\frac{\sum_n P^{old}(j|x^n)\|x^n - \mu_j^{new}\|^2}{\sum_n P^{old}(j|x^n)} \tag{8}$$

is applied. Finally, to update the prior probabilities we set:

$$P(j)^{new} = \frac{1}{N}\sum_n P^{old}(j|x^n) \tag{9}$$

Our video abstraction algorithm proceeds as follows. Initially, the number of principal modes used, d, is chosen such that at least 97.5% of the variance of the clip is accounted for. A $(d+1)$ dimensional vector is formed by the response of each image to these modes plus the frame number. Each image is now a point in a low-dimensional eigen-time space. An initial GMM consisting of two components is then trained to fit the data. How well the GMM models the data is computed and compared to a threshold, such that $\max(-\log(p(x^n))) < t$, where $p(x^n)$ is the likelihood of each data point given the model. If the threshold has been reached by the current model then a good approximation of the data has been found. If this is the case then the frame nearest to each cluster centre (in a maximum likelihood sense) is selected and these form the key frames describing the clip. These key frames are time ordered to produce the final abstraction.

If the threshold has not been reached then the number of components in the GMM, M, is incremented. This process is iterated until the threshold is reached or until the maximum number of GMM components is exceeded. The maximum number of components has been selected by the editors so that when the key frames are displayed next to each other their combined width does not exceed the width of the screen. In the examples shown here, $2 \leq M \leq 12$. The complexity of the EM algorithm ensures that random restarts are necessary to ensure that 'simpler' GMMs are not missed, and that if a fully covariant model completely fails a diagonal covariance is occasionally necessary.

For our database, the number of principal modes needed to account for 97.5% of the variance in a clip is in the range $1 \leq d \leq 6$. For clips that contain little change in frame content and no shot changes the first mode alone is sometimes enough to account for 97.5% of the variance.

3. Results

The technique described above was applied to our database of 307 clips of wildlife footage. The summaries give an

immediate and visually appealing overview of the entire database.

Figure 4 shows the result of clustering the eigenspace

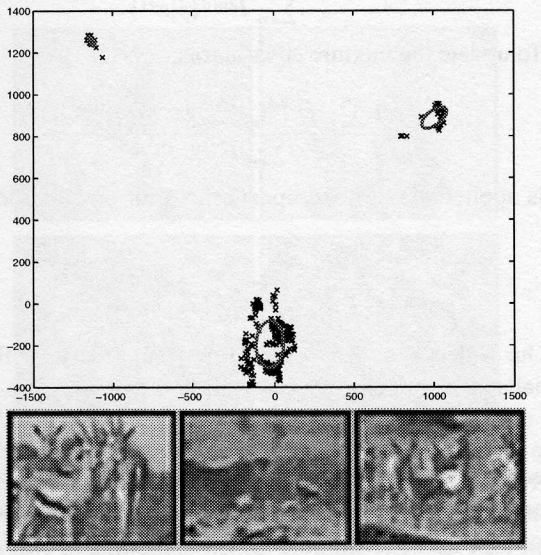

Figure 4: Clustering the eigenspace described in Figure 2. Three clusters are obtained (top), each one of which has a corresponding representative key frame (bottom). A threshold of $t = 15$ was used.

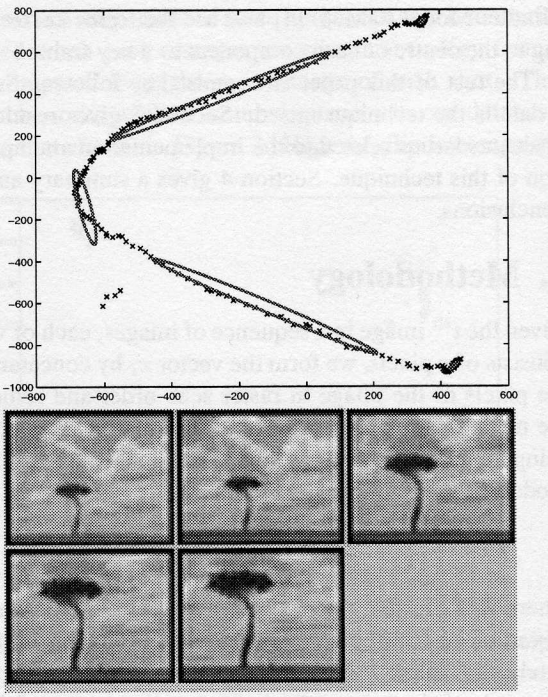

Figure 5: Clustering the eigenspace described in Figure 3. Five clusters are obtained (top), each one of which has a corresponding representative key frame (bottom). A threshold of $t = 15$ was used.

first described in Figure 2. As expected, three clusters are found, and their corresponding key frames are shown. Similarly, Figure 5 shows the five key frames obtained when clustering the sequence described in Figure 3, with the same maximum likelihood threshold of $t = 15$.

The only user interaction required in this process is the setting of the threshold, t, which determines how well the GMM models the data. To aid in this process, Figure 6 shows a plot of the maximum likelihood threshold value against the average number of frames selected for each clip in the database. Given this graph, an unskilled operator is able to control the average number of key frames selected for each clip in the database. The threshold may also be automatically adjusted so as to obtain a specified number of frames for each clip.

Setting the threshold lower will generally increase the number of Gaussians required to model the data and hence the number of key frames used to represent each clip. Figure 7 shows the result of setting $t = 13$. In this case the GMM tends towards a line approximation of the data, and the 5 outlying frames at the end of the clip (the cluster in the bottom left-hand corner) now have their own key frame assigned.

Another, similar example, is given in Figure 8. Here the clip contains a barely visible orang-utan, swinging through a canopy of trees, with some colourbars at the end. The abstraction is a good representation since little motion occurs in the main body of the clip.

In the above Figures, frame number (time) has been removed to aid clarity, even though time is included as one of the dimensions to be clustered. It is clear that this feature aids the clustering process since, as is shown in Table 1, the average number of key frames abstracted per clip decreases dramatically when time is used. The dimensionality of the space used to cluster each clip remains constant, but using time as one of the dimensions results in a decrease in video abstraction length of around 20%. In Figure 9, a more complicated clip is abstracted. Here, we show a Sammon mapping [11] of the 3D space (2 eigenvalues plus

Threshold $t = 13$	Time	No time
Average clip length	7.02	8.87

Table 1: Result of video abstraction for a fixed threshold t, with and without frame number (time) used as part of the space to be clustered. The use of time significantly reduces the number of Gaussians required to successfully cluster the feature space.

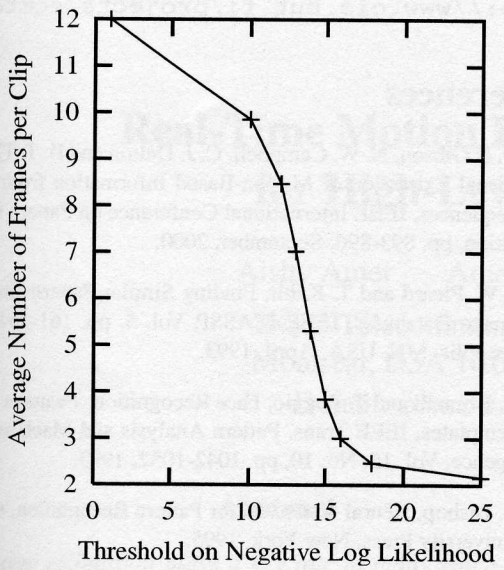

Figure 6: Plot of different threshold values against number of frames chosen.

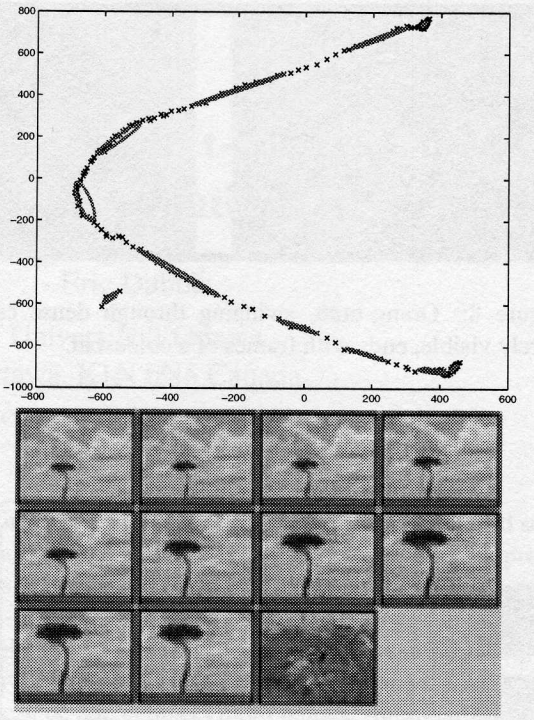

Figure 7: The video abstraction of the tree zoom sequence with a threshold of $t = 13$. In this case 11 key frames are abstracted, including one of the 5 outlying frames accidentally left on the end of the sequence when it was digitised.

frame number) onto a 2D space. The GMM selected and the key frames obtained are also shown. In this clip a number of events occur. Initially a hyena is walking towards the camera in close up. This is then followed by a dissolve to a long shot of a beast walking in the distance and ends by a dissolve to a close up of another beast's legs. This extremely complex clip is abstracted well using our approach despite the multiple dissolves. If frame number is not used the space is difficult to cluster meaningfully and a poor abstraction results.

Perhaps the most complex clip in our database is abstracted in Figure 10. This is a trailer compiled for a television show, containing many post-production effects such as shot changes, fades and dissolves. Once again our approach clusters the space well and produces a representative abstraction containing 11 frames from the original 853 frame clip.

4. Summary and Conclusions

In this paper, we have demonstrated a novel approach for clip-based key frame abstraction on a large database of wildlife video. We have shown that creating key frame video abstractions can be achieved by transforming each frame of a video sequence into an eigenspace and then clustering this space using Gaussian Mixture Models. An iterative process computes a GMM configuration that best clusters the data based on a maximum likelihood threshold. The images nearest to the centres of each of the GMM components are selected as key frames. This ensures that representative frames are obtained, rather than frames at, or very near, a shot change.

Unlike previous work this technique relies on global video clip properties and results show that the key frames extracted give a good representation of the overall clip content. Our approach can cope with clips containing very subtle motions just as well, and within the same framework, as those containing rapid shot changes, fades and dissolves.

We have shown that the use of frame number (time) as part of the clusters process allows the eigenspace to be clustered more easily and leads to better abstractions. We have also demonstrated that, for our large database of wildlife clips, the average length of the abstraction can be easily controlled using a single user-defined parameter.

Acknowledgements

This work is sponsored by the DTI Broadcast Link Programme. Thanks are due to British Film & Video and the BBC Natural History Unit (Bristol).
The GMM code used here is based in part on Netlab `http://www.ncrg.aston.ac.uk/netlab`
Sammon mapping code came from the Sombox toolkit

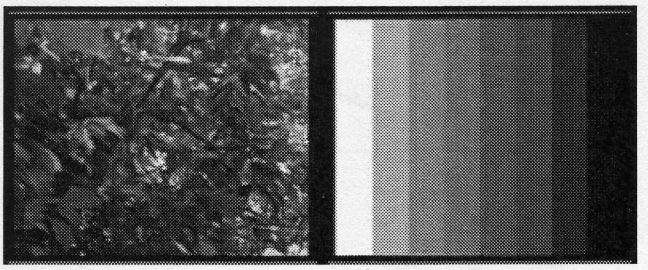

Figure 8: Orang-utan, swinging through dense canopy, barely visible, ends with frames of a colourbar.

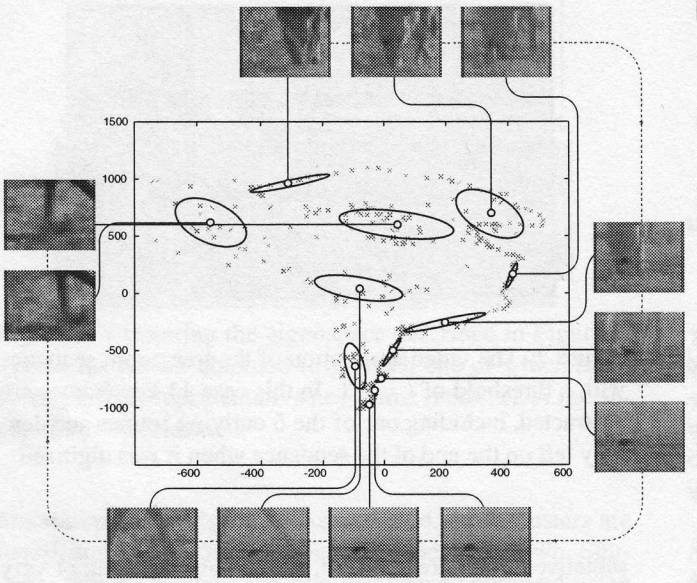

Figure 9: A 2D Sammon mapping of 3D space (two eigenvalues plus time) with the observed clusters superimposed. Key frames corresponding to each cluster are shown around the edges.

Figure 10: Abstraction of a trailer for a television wildlife programme. It contains many different shots, cuts and dissolves.

http://www.cis.hut.fi/projects/somtoolbo

References

[1] D. P. Gibson, N. W. Campbell, C. J. Dalton and B. T. Thomas, Visual Extraction of Motion-Based Information from Image Sequences, IEEE International Conference on Pattern Recognition, pp. 893-896, September, 2000.

[2] R.W. Picard and T. Kabir, Finding Similar Patterns in Large Image Databases, IEEE ICASSP, Vol. 5, pp. 161-164, Minneapolis, MN, USA, April, 1993.

[3] R. Brunelli and T. Poggio, Face Recognition: Features versus Templates, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 15, No. 10, pp. 1042-1052, 1993.

[4] C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, New York, 1995.

[5] S. V. Porter, M. Mirmehdi and B. T. Thomas, Video Cut Detection using Frequency Domain Correlation, IEEE International Conference on Pattern Recognition, pp. 413-416, September, 2000.

[6] B. Gunsel and A. M. Tekalp, Content-Based Video Abstraction, IEEE International Conference on Image Processing, Chicago, Il, Oct., 1998.

[7] C. Kim and J-N. Hwang, An Integrated Scheme for Object-based Video Abstraction, Proceedings of the 8th International Conference on Multimedia, ACM Press, New York, pp. 303–312, Oct., 2000.

[8] M. Smith and T. Kanade, Video Skimming and Characterisation through the Combination of Image and Language Understanding, IEEE International Workshop on Content-Based Access of Image and Video Databases, pp. 61-70, Jan., 1998.

[9] E. Sahouria and A. Zakhor, Content Analysis of Video Using Principal Components, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 8, pp. 1290-1298, Dec., 1999.

[10] K. J. Han and A. H. Tewfik, Eigen-Image Based Video Segmentation and Indexing, IEEE International Conference on Image Processing, pp. 147-151, Oct., 1999.

[11] J. W. Sammon Jr., A Nonlinear Mapping for Data Structure Analysis, IEEE Transactions on Computers, Vol. C-15, No. 5, pp 401-408, May, 1969.