A Fast Method for Textual Annotation of Compressed Video

Amit Jain and Subhasis Chaudhuri
Department of Electrical Engineering
Indian Institute of Technology, Bombay,
Mumbai - 400076. INDIA.
{ajain,sc}@ee.iitb.ac.in

Abstract

We present a method for automatic textual annotation of compressed video. Initially, cuts are detected in the video sequence where consecutive frames show a large difference. Key frames are then extracted to represent each shot. A video sequence is, thus, condensed into a few images, hence this forms a compact key frame representation of the video. Each key frame is compared with an annotated database of images, using standard CBIR techniques, to obtain a textual description of the scene. The entire process is designed to work on MPEG-DC sequences, where only a partial decoding is required for performing content-based operations on MPEG compressed video streams.

1. Introduction

In recent years, technology has reached a level where a vast amount of digital information is available at a low price. The ease and low cost of obtaining and storing digital information as well as the almost unlimited possibility to manipulate it makes it a popular means of storing data. One of the key features required in a visual information system is efficient indexing to enable meaningful and fast classification of objects in a video database.

It is generally accepted that content analysis of video sequences requires a preprocessing procedure that first breaks up the sequences into temporally homogenous segments called *shots* [1],[5],[7],[9], [12]. A shot is a sequence of frames generated during a continuous operation and therefore represents continuous action in time or space. Shots represent temporal segments of a video with smoothly changing contents. These segments are condensed into one or a few representative frames (key frames) [2] [13] to yield a pictorial summary of the video. Semantic video indexing based on object segmentation is discussed in [8], [14]. Object and camera motion can also be used for analyzing and annotating video [3],[11]. Video annotation is often facilitated by prior knowledge of some general structure for the class of video under study. A basketball annotation system is discussed in [11].

Since most of the video streams that modern digital storage systems have to deal with are available in the MPEG compressed format, no content-related operations are possible on these streams directly. The analysis of compressed video can proceed in one of the two fundamental ways. The first is by decompressing the video bitstream and using the individual frames to gather information about various characteristics of the video such as content or motion, and extracting indexable features in the pixel domain. Operations on fully decompressed video do not permit rapid processing because of the data size. The second method involves exploiting encoded information contained in the compressed representation. This method operates directly on a small fraction of the compressed data. Such greatly reduced data is readily extracted from compressed video without full frame decompression and still captures the essential information of the video. The information available in compressed representation of video includes the types of each Macroblock (MB), the Discrete Cosine Transform (DCT) coefficients of each MB. and the motion vector components for the forward, backward and bidirectionally predicted MBs.

In this paper, we present a computationally efficient method for automatic annotation of video sequences. The approach makes use of the compressed domain video data and can be split into two parts - segmentation and indexing. Temporal segmentation divides the video into shots or scenes. The segmentation is done by partially decoding the compressed video bitstream to obtain DC sequences. By comparing the subsampled frames, a scene change is detected where successive frames show a large difference. Indexing is done by choosing key frames to represent each scene. This gives a compact pictorial representation of the contents of the video. By comparing the key frames to a database of annotated still images, using standard CBIR techniques, a textual annotation of the scene is obtained. Hence, a textual representation of the complete video is now available.

The organization of the paper is as follows. In Section 2, we briefly discuss the generation of DC-Image and DC-

Figure 1: Full Image at 352 x 240 and its DC image at 44 x 30.

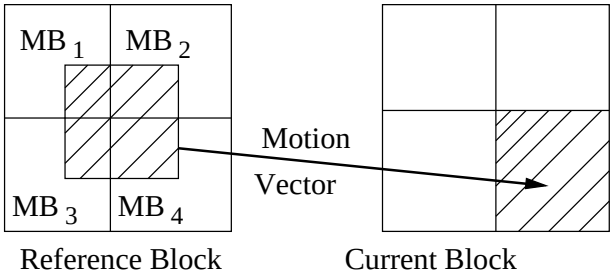


Figure 2: Illustration of how the DC coefficients for a macro block is computed in a P frame.

Sequence. Section 3 addresses the problem of temporal segmentation of the video. Section 4 deals with key frame selection and generation of textual index of the video. The results are presented in Section 5 followed by conclusion in Section 6.

2. DC Image and DC Sequences

DC images are spatially reduced (one pixel per macroblock) versions of the original images. The DC image is nothing but a top level representation of an image in its multiresolution pyramid. Such spatially reduced images, once extracted, are very useful for efficient scene change detection and other applications [12]. In this section we briefly discuss how the DC-image and the DC-sequence can be efficiently extracted from compressed videos [10].

We focus on MPEG-2 video streams. We restrict ourselves to using data which can be easily extracted from MPEG bit streams without the full frame decompression. Specifically, we use the frame number, frame encoding type (I, P, or B), and the DC coefficient of each DCT-encoded pixel block.

The (i, j) pixel of the DC-image is the average value of the $(\mathrm{i,j})^{th}$ macroblock of the original image. While the DC-image is much smaller than the original image, it still retains significant amount of information. The subsampled video sequence formed in such manner is called the *DC sequence*. This entails significant savings in computational and storage costs, resulting in a faster annotation. Fig. 1 illustrates an original image of size 352 x 240 and its DC-image of size 44 x 30 using N = 8.

The extraction of DC image from I-frame in MPEG is trivial since the DCT coefficients are readily accessible for I-frames. For the I-frames of MPEG, the original image is grouped into 8 x 8 blocks, and the DC term c(0,0) of its 2-D DCT is related to the pixel values f(i,j) via

$$c(0,0) = \frac{1}{8}\sum_{i=0}^{i=7}\sum_{j=0}^{j=7} f(i,j)$$

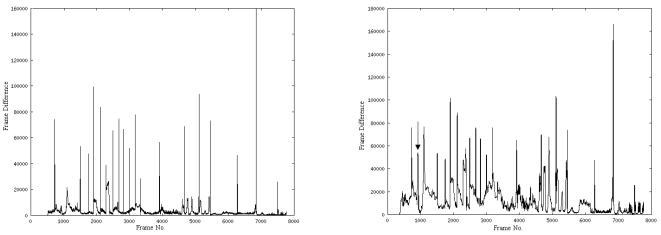
which is 8 times the average intensity of the block.

Since P and B-frames are represented by the residual error after prediction or interpolation, their DCT coefficients need to be estimated. To calculate the DCT coefficients of a Macroblock (MB) in a P frame or B-frame, the DCT coefficients of the 8 x 8 area of the reference frame that the current MB was predicted from need to be calculated. Suppose this area is called the reference MB (though it is not an actual MB). Since the DCT is a linear transform, the DCT coefficients of the reference MB in the reference frame can be calculated from the DCT coefficients of the four MBs that can overlap this reference MB, albeit with substantial computational expense. It is easy, however, to calculate reasonable approximations to the DC coefficients of an MB of a P or a B-frame. Fig. 2 shows an MB in a P-frame, MB_{Cur}, being predicted from a 8 x 8 area denoted by MB_{Ref}. While encoding the P-frame, only the residual error of MB_{Cur} with respect to MB_{Ref} is stored. The DC coefficients of MB_{Ref} can be calculated from the DCT coefficients of four MBs- MB_1, MB_2, MB_3 and MB_4. To avoid expensive computation, the DC coefficient alone is approximated by a weighted sum of the DC coefficients of the four MBs, with the weights being the fractions of the areas of these MBs that overlap the reference MB, i.e,

$$DC(MB_{Ref}) = \sum_{i=1}^{4} w_i.DC(MB_i)$$

where w_i is given by the ratio of the area of the shaded region of MB_i to its total area. Similarly, if an MB in a B-frame is interpolated from two reference MBs, its DC coefficient is approximated by an average of the estimated DC coefficients of each of these two MBs.

The partial decoding of a MPEG video bitstream, using only the DC term of the DCT, gives a spatially decimated version of the video. This is called the DC sequence. The frames in a DC sequence are spatially reduced by a factor of 8 in both the dimensions. In our proposed scheme, we form the DC sequence of the video by operations only on the luminance channel which leads to significant savings in computational and storage costs.



a. Abrupt b. Gradual

Figure 3: Illustration of Interframe difference D_p^q for a DC sequence of the movie for (a) abrupt scene change (q=1), (b) gradual scene change (q=10).

3. Temporal Segmentation

To analyze the content of a given video sequence, we need to first segment the sequence into individual shots. The DC sequences extracted from compressed video are used to perform temporal segmentation of the video. As discussed in [12], two separate detection algorithms are used to detect abrupt and gradual scene changes, respectively. We further improve the shot detection method to handle transmission disturbances. The method is based on pixel level differences between DC images. Let $X = x_{i,j}$ and $Y = y_{i,j}$ be two DC images and let $d(X, Y)$ denote their difference. The difference metric is set as pixel level difference, i.e.

$$d(X, Y) = \sum_{i,j} |x_{i,j} - y_{i,j}|.$$

3.1 Abrupt Scene Changes

To detect abrupt scene changes, peaks are detected in the plot of the pixel level differences of successive frames in a DC sequence. Since scene change is a local activity in the temporal domain, a threshold to detect the peak is set to match the local activity. A $sliding\ window$ is used to examine m successive frame differences. Let $f_k, k = 1, 2, ..., N$ be a sequence of DC images. A difference sequence $D_k = d(f_k, f_{k+1})$ is formed. A scene change is declared from f_l to f_{l+1} if

i. $D_l = max\ (D_{l-m}, ..., D_{l+m})$, and
ii. $D_l \geq \alpha . D_l'$

where D_l' is the next largest value within the same window $[-m, m]$.

The condition (i) detects the actual peaks and the condition (ii) guards against fast panning or zooming scenes. The window size m is set to be smaller than the minimum duration between two consecutive scene changes. It has been found that values of α ranging from 3.0-4.0 give good results. Fig. 3 (a) shows the plot of interframe pixel level difference applied to DC sequence of an MPEG bitstream.

3.2 Gradual Scene Changes

A robust way of detecting gradual transition is by using not a single frame difference D_k, defined earlier, but the q^{th} frame difference D_k^q in the DC sequence where D_k^q is defined as

$$D_k^q = d(f_k, f_{k+q}).$$

By selecting q greater than the duration of the transitions, we get "plateaus" in the plot of D_k^q. One now needs to detect such plateaus in the plot D_k^q. We use the following criteria to detect a plateau. We declare the start of a plateau at frame $k = l$ (i.e, start of a gradual scene change) if

i. $|D_l^q - D_{l+j}^q| \leq \epsilon_1$ for $j = 1, 2, .., q$ and
ii. $|D_l^q - D_{l-1}^q| > \epsilon_2$

where ϵ_1 is a threshold that allows small variations in the height of the plateau, ϵ_2 is a large threshold (usually $\epsilon_2 \gg \epsilon_1$) demands that the plateau rises quite steeply when the gradual transition starts, and the parameter q (as explained earlier) defines the minimum dissolve or fade out duration. Fig. 3 (b) shows the plot of D_p^q for q=10. The arrow in Fig. 3 (b) points to a "plateau".

3.3 Scene Changes in Presence of Transmission Disturbances

Large differences between the DC images may be encountered because of the transmission disturbances. This may happen due to bursty network congestion when a number of packets may get dropped during the streaming process. We use the observation that such disturbances manifest themselves as two sharp peaks quite close to each other and intersparsed with fairly large values in the difference plot of the DC images. The two peaks correspond to the start of the burst error and the resumption of the quality transmission, respectively. Ideally these should not be considered as scene changes. We solve this problem by neglecting whenever there are two peaks in the difference plots close to each other. It should be noted that during the normal transmission, there will be occasional packet drops but the increase in the difference metric is quite marginal. The conditions given in section 3.1 are able to handle such a situation. A scheme as simple as described above is found to provide enough resilience to transmission disturbances.

3.4 Overall Scene Change Detection

To detect scene changes in a video, we combine both the algorithms discussed earlier to detect both abrupt and gradual scene changes. We keep in mind that after any scene change, a gradual scene change cannot take place before a particular number (n_0 where $n_0 > m$) of frames. Hence if a scene change is declared at a particular frame, then we

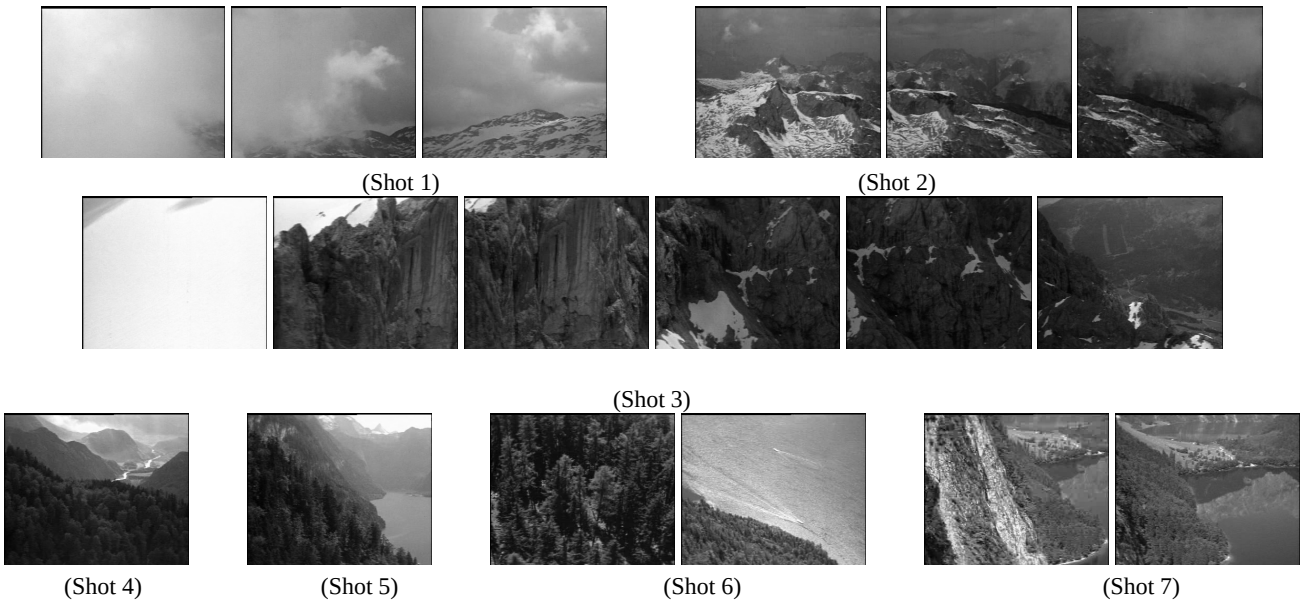


Figure 4: Pictorial summarization of the movie through key frame selection in each shot.

look for the gradual scene change n_0 frames after the abrupt scene change. However, there could be another abrupt scene change and hence the scheme restricts the computation to searching only an abrupt scene change immediately after a detected shot. We also include the modification suggested in section 3.3 to robustly detect the shots when transmission disturbances are expected.

Each shot, thus obtained, forms the basis for further analysis, that is, selection of key frames.

4. Indexing

The approach to indexing a video involves extracting a set of key frames for the entire video clip, to capture as much of the contents of the video as possible, but at the same time redundant frames are excluded. To create a pictorial summary of the video, representative images have to be selected from each shot. The representative frame for each shot can be selected from amongst the frames in the middle to avoid special effects such as fade-in and fade-out which are more often found at the beginning or end of a shot. This gives a crude pictorial summary of the video since the use of only one frame per shot is insufficient to capture the time varying essential features of video sequences. Therefore, there could be multiple key frames in the same shot.

We use the DC histogram comparison technique [4] for the *nonuniform* sampling of all frames that constitute a shot. The proposed algorithm for key frame selection is based on DC coefficients that are calculated only for the Y (luminance) component. This is because (1) human visual system is more sensitive to Y component than to the other chrominance components, and (2) the MPEG standards typically use higher spatial sampling for Y than the same for the other two components. The histograms of DC coefficients of the Y component of all the *I*-frames constituting a shot are compared. Further, to reduce the computation, we use only the DC-image (the thumbnail image) for all analyses. The similarity metric used to compare the histogram of the i^{th} and j^{th} DC images is $L(i,j)$ metric, defined as:

$$L(i,j) = (\sum_{k=1}^{k=N} [H_i(k) - H_j(k)]^2)^{\frac{1}{2}}$$

where $H_i(k)$ denotes the k^{th} histogram bin value of the of i^{th} DC image. We declare the i^{th} frame to be a key frame if it is sufficiently different from the previous key frame, i.e. if

$$L(i, KF) \geq \delta \quad (1)$$

and

$$L(i+1, KF) \leq L(i, KF) \quad (2)$$

Here δ is a threshold that controls the density of temporal sampling and KF denotes the last key frame detected in the sequence. The second condition ensures that the chosen key frame is maximally apart from the previous KF compared to the rest of the candidate frames.

The described process is applied to I-frames only since two consecutive I-frames have a larger interframe difference than two consecutive P or B-frames. Hence I-frames are the best candidates to capture the temporal variations in

Table 1: Textual Summary of the Movie Segment “Sound of Music”.

Shot No.	Frames	Key Frames	Key Frame Annotation	Shot Annotation
1	0500-0725	0500	Snow covered Slope	Mountain Peak
		0560	Mountain Peak	
		0694	Mountain Peak	
2	0726-0949	0726	Cliff	Cliff
		0787	Cliff	
		0834	Cliff	
3	0950-1501	0950	Snow covered Slope	Cliff
		1185	Cliff	
		1247	Cliff	
		1309	Cliff	
		1394	Cliff	
		1462	Cliff	
4	1502-1748	1502	River	River
5	1749-1901	1749	Gorge	Gorge
6	1902-2110	1902	Forest	Forest
		2098	Forest	
7	2111-2286	2111	Gorge	Gorge
		2217	Gorge	

a shot as it requires the minimal decompression. The first I-frame in a video clip (shot) is always declared as the key frame. Then the other frames are compared to this frame. We declare the i^{th} frame to be the new key frame if it satisfies conditions (1) and (2). Criterion (1) is to ensure that the current frame is significantly different from the previously declared key frame. Criterion (2) is imposed to ensure that the frame which is most different from the current key frame is selected as the new key frame. The subsequent frames are then compared to this new key frame.

It is appropriate to set the threshold locally, since the number of key frames required to represent a scene depends on the local activity in the scene. The proposed method sets the threshold to be β times the standard deviation of the DC image of the preceding key frame. It has been experimentally found that values of β ranging from 2.5 - 4.5 give good results.

To obtain the textual annotation of the video each key frame from every shot is compared with a database of textually annotated still images, using any standard content based image retrieval (CBIR) technique. Any CBIR technique can be used, in principle, and we use the CBIR method proposed in [6]. We form the image database by key framing a large number of video sequences and manually annotating these key frames. Now the best match for a given key frame from a test video sequence is obtained. The textual annotation of the best match image from the database highlights the content of a key frame and gives a textual description of the scene. Thus a sparse but coherent and temporally ordered textual description of the video is generated.

5. Results

To test the proposed algorithm on realistic, compressed video, we have applied the algorithm to a wide variety of MPEG video test sequences. All video clips are digitized at 25 frames/sec and at a resolution of 352x288 pixels. All the images presented in this section are converted to black and white for the purpose of presenting results in the paper. For the performance analysis of the proposed scheme, the ground truth was obtained through human intervention. The database used for comparing the key frames is obtained by manually annotating each image present in the database.

The result for the entire scheme of annotating a video is presented for clip from the movie “Sound of Music” and is of 5 minutes 11 seconds duration. It is mostly about the panning of a camera around a natural surrounding before the lead actress enters the scene. We neglect frames 0-499 because they contain captions. Fig. 3 shows the plot of the frame difference for the DC sequence obtained from the movie clip and Fig. 4 depicts the pictorial summary of the movie upto the 7th shot due to the brevity of the presentation. The textual summary of the video sequence is presented in table I. For scenes which generate multiple key frames, we define the shot annotation to be the most occurring key frame annotation. The first shot is represented by

three key frames. Out of these three key frames, the first key frame is annotated as “Snow covered Slope” whereas the remaining two key frames are annotated as “Mountain Peak”. The first key frame is actually an image of a cloudy sky, but the CBIR technique annotates it to ”Snow covered Slope”. We select the shot annotation as the most occurring key frame annotation, i.e. “Mountain Peak”. The second shot is represented by three key frames and since each key frame is annotated as “Cliff”, the scene annotation is also “Cliff”. The third shot has six key frames, out of which the first is annotated as “Snow Covered Slope” and the rest are annotated as “Cliff”. Hence the shot annotation is “Cliff”. The next shot is annotated as “River” since the key frame which represent the shot is annotated as “River”. The fifth shot is depicted by a single key frame which is annotated as “Gorge” and the shot annotation is the same. The sixth shot is annotated as “Forest”, since both the key frames are annotated as “Forest”. The last scene has only one key frame which is annotated as “Gorge”.

The efficacy of the proposed algorithm depends on the scheme used for temporal segmentation. The scheme implemented for temporal segmentation gives a detection rate of 95% and precision rate of 92%. The ability of the key frames to highlight the temporal variations in a shot is, however, a subjective issue. The effectiveness of the key frame selection algorithm is also tested using the ground truth data obtained manually. It is observed that for the movie clip, 18 frames are declared as key frame candidates; while there are 2 additional frames which could have been declared as possible key frame candidates. Hence the accuracy in key frame selection is 90%. It is needless to mention that quality of textual annotation depends on the effectiveness of the CBIR algorithm and how exhaustive is the annotated database.

6. Conclusions

We present in this paper the concept of video visualization in the form of pictorial and textual summary generation. The pictorial summary created by key frame selection compactly represents the original video with considerable fidelity. The temporal progress of the story is preserved by presenting the key frames in the time order. A technique has been proposed to create a textual summary of the video. The textual and pictorial summary contribute to semantic visualization and succinct presentation of the pictorial content of video.

All of the key frames based approaches represent a video by a small set of images. Thus, they lose the motion information of the original video. The proposed algorithm also suffers from the same defect. By the additional use of motion information of the video, better indexing and retrieval techniques can be developed, which is our future research problem.

References

[1] P. Bouthemy, M. Gelgon, and F. Ganasia. A Unified approach to Shot Change Detection and Camera Motion Characterization. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1030 – 1044, October 1979.

[2] H. S. Chang and S. U. Lee. Efficient Video indexing Scheme for Content Based Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8):1269 – 1279, December 1999.

[3] S. Dagtas, W. Al-Khatib, A. Ghafoor, and R. L. Kashyap. Models for Motion-based Video Indexing and Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(1):88 – 101, January 2000.

[4] B. Furht and P. Saksobhavivat. A Fast Content-Based Multimedia Retrieval Technique Using Compressed Data. `www.cs.fau.edu/`$borko/paper\_SPIE-1.pdf$.

[5] U. Gargi, R. Kasturi, and S. H. Strayer. Performance Characterization of Video-shot-change Detection Methods. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1):1 – 13, February 2000.

[6] N. Jhawar, S. Chaudhuri, G. Seetharaman, and B. Zavidovique. Content Based Image Retrieval Using Optimum Peano Scan. Proc. Intl. Conf.Pattern Recognition (ICPR), Quebec City, Canada, August 2002.

[7] S.-W. Lee, Y.-M. Kim, and S. W. Choi. Fast Scene Change Detection Using Direct Feature Extraction from mpeg Compressed Videos. *IEEE Transactions on Multimedia*, 2(4):240 – 254, December 2000.

[8] M. R. Naphade and T. S. Huang. A Probabilistic Framework for Semantic Video Indexing, Filtering and Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(1):141 – 151, March 2001.

[9] S.-C. Pei and Y.-Z. Chou. Efficient mpeg Compressed Video Analysis Using Macroblock Type INformation. *IEEE Transactions on Multimedia*, 4(1):321 – 333, December 1999.

[10] J. Song and B.-L. Yeo. Fast Extraction of Spatially Reduced Image Sequences from MPEG-2 Compressed Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1100 – 1114, October 1999.

[11] Y.-P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge. Rapid Estimation of Camera Motion from Compressed Video with Application to Video Annotation. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1):133 – 146, February 2000.

[12] B.-L. Yeo and B. Liu. Rapid Scene Change Analysis on Compressed Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6):533 – 544, December 1995.

[13] M. M. Yeung and B.-L. Yeo. Video Visualisation for Compact Presentation and Fast Browsing of Pictorial Content. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(5):771 – 785, October 1997.

[14] D. Zhong and S.-F. Chang. An Integrated Approach for Content-Based Video Object Segmentation and Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8):1259 – 1269, December 1999.