A Framework for Sub-Window Shot Detection

Chuohao Yeo[1], Yong-Wei Zhu[2], Qibin Sun[3], Shih-Fu Chang[4]
[1,2,3]*Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore*
[4]*Department of Electrical Engineering, Columbia University*
{zuohao}@eecs.berkeley.edu, {ywzhu,qibin}@i2r.a-star.edu.sg, {sfchang}@ee.columbia.edu

Abstract

Browsing a digital video library can be very tedious especially with an ever expanding collection of multimedia material. We present a novel framework for extracting sub-window shots from MPEG encoded news video with the expectation that this will be another tool that can be used by retrieval systems. Sub-windows shots are also useful for tying in relevant material from multiple video sources. The system makes use of Macroblock parameters to extract visual features, which are then combined to identify possible sub-windows in individual frames. The identified sub-widows are then filtered by a non-linear Spatial-Temporal filter to produce sub-window shots. By working only on compressed domain information, this system avoids full frame decoding of MPEG sequences and hence achieves high speeds of up to 11 times real time.

1. Introduction

With the advent of cheaper and faster processing power and storage, there has been a wide-spread proliferation of digital multimedia material, including digital video. This has unfortunately meant that humans are increasingly being overwhelmed by digital content, and there is a need to be able to get relevant information quickly.

Most digital video retrieval systems use shots as the basic unit in analyzing a video for browsing [1,2]. To this end, much work has already been reported in the literature for shot boundary detection. Shot analysis has been done in both the uncompressed domain and on MPEG compressed videos [3]. More recent work have focused on MPEG videos since they are very wide-spread and lend themselves easily to fast shot analysis. Parameters such as coding bit-rate [4], ratio of forward/backward references in B-frames [4-10], number of Intra-coded MacroBlocks (MBs) in P-frames [9] can be analyzed for shot boundaries. In addition, statistics such as Discrete-Cosine-Transform (DCT) DC variances [5,9,10] and DCT DC images [3] have also been used.

However, there has been no work reported on the detection of sub-window shots. In this paper, we define a sub-window shot as a rectangular area of interest that presents visually distinct material from its surroundings and is semantically independent from the rest of the frame. This is widely referred to as a "picture-in-picture" effect. Such shots frequently appear in news programs where the news anchor is either introducing another scene or presenting a field correspondent. Figure 1 shows an example of such a sub-window shot.

Figure 1. An example of a sub-window shot

Since such sub-windows may provide extra segmentation within the same news scene, it would be useful to include them for the purpose of news story segmentation [19]. In addition, it allows for more relevant comparison of visual material from multiple news video when performing video browsing [20].

This paper proposes a basic framework for the detection of rectangular sub-window shots in MPEG [11] domain. Only MPEG-1 [12] compression parameters will be used to ensure backward compatibility so that the framework described can be applied widely. The MPEG Development Classes

implemented by Dongge et al. [13] was used to extract MPEG parameters needed in the analysis.

Section 2 introduces the general outline of this framework. Section 3 briefly describes each of the main processing blocks. Experimental results are described and discussed in Section 4. Section 5 concludes the paper.

2. Framework outline and rationale

Because the system is designed to work wholly in the MPEG domain with minimal decoding, the sub-windows are found only to MB accuracy. We believe that this is sufficient for video browsing since there is no real need for pixel-accurate sub-windows. In addition, such boundaries lend themselves easily for trans-coding.

Figure 2 shows the block diagram for the processing.

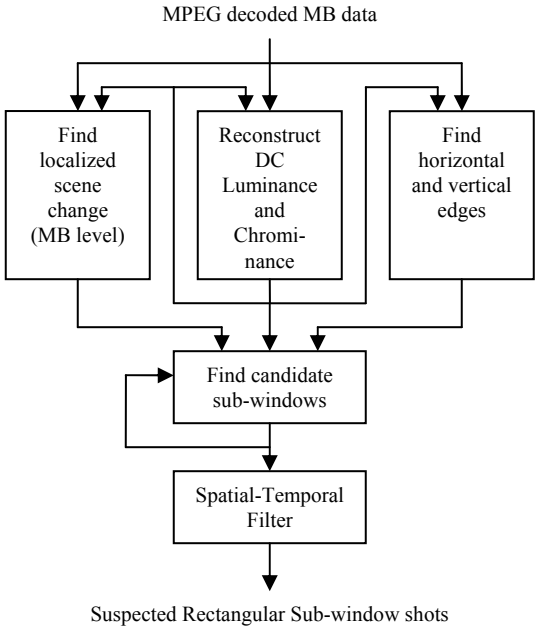


Figure 2. Block diagram of MPEG Sub-window shot detection

The input is the decoded MB information. In particular, the only parameters used are MB coding type, the reconstructed DCT coefficients, and the motion vectors. Hence, there is no need for the MPEG video to be fully decoded, and this allows for fast sub-window shot detection.

In this framework, the output is the location of the sub-window shot in space and time. It would be trivial to have the output as either separate MPEGs or key-frames if desired.

Essentially, the system makes use of local scene change at the MB level to first detect the possible presence of sub-windows. The detection is further refined using other features based on the following observations:

- Sub-windows usually have very distinct borders. Hence, edge information plays a very important part in determining if an area of local scene change is actually part of a sub-window or otherwise.
- Sub-windows tend to have different color properties from areas outside it. Hence, the chrominance statistics of a suspected sub-window and the area outside it would be important.
- Sub-windows tend to have coherent borders in terms of chrominance. As above, the chrominance statistics of the border area of a suspected sub-window would be important.
- Sub-window shots last for a significant amount of time (>2s) and are stationary in space. Therefore, spatial-temporal filtering is used to filter out spurious occurrences and to smooth out its spatial extent.

3. Main processing blocks

The framework depends on several distinct processing blocks to achieve the desired output. By organizing the system this way, it is possible to improve the framework rapidly by refining any of the processing blocks.

3.1. Reconstruct DC luminance and chrominance images

The DC Luminance image of the frame, with the same dimensions as that of the block layer, is reconstructed using a first-order approximation as described by Yeo et al. [3]. Because chrominance information will also be necessary, the DC Chrominance images are also reconstructed in the same way, but with the dimensions as that of the MB layer.

3.2. Find localized scene change

From the work reported in the literature on shot boundary detection [3-10], we derived the following rules for deciding that a MB is part of a local scene change:

- If a MB is Intra coded, and the DC Luminance change from the MB of the same location in the forward referenced frame is greater than δ_{DC}.

- If a MB is forward-predicted or bi-directionally-predicted, and the DC term of the correction in Luminance is greater than δ_{DC}.
- If the current frame is a B-frame, and the MB is only backward-predicted, and the DC Luminance change from the MB of the same location in the forward referenced frame is greater than δ_{DC}.

δ_{DC} is a threshold that is set to the minimum change in average intensity of a MB when its content changes. The set of localized scene change MBs produced at each frame *n* is denoted by *L(n)*.

This processing block produces many spurious localized scene change MBs, since it could be due to both actual local scene change as well as other un-translational transformations that cannot be efficiently coded using motion compensation. It may also be encoder dependent, since the coding decision for MBs in predicted frames is based largely on the need to reduce bit-rate and not necessarily on content change or localized movement. The DC Luminance value may also not be a good indication of the content of the area.

Despite its many shortcomings, this block need only produce a coarse estimate of where localized scene change might be. The spatial-temporal filter at the last stage of the processing filters out most noise that originates here.

3.3. Horizontal and vertical edge detection

Accurate edge detection is the most important part of the whole process because the crucial difference between an actual sub-window shot and false positives has been observed to be the existence of a well-defined border around a region of localized scene change. In particular, we need to be able to detect horizontal and vertical edges accurately since rectangular sub-windows are to be detected.

The key problem here is detecting edges without uncompressing the MPEG; hence, only MB and block information are available. Having recognized this, there are two types of edges to be detected and they require different methods of detection. These are edges that occur on the boundaries of MB or blocks, and edges that occur within MB or blocks.

Edges that occur on the boundaries can be detected using any textbook edge detection method on the DC Luminance and Chrominance images. The DC Chrominance images are used because in many cases, sub-windows in Chrominance images actually appear more distinct than in Luminance images. For our implementation, we used a zero-crossing detection of the second derivative approximated by a 3x3 8-neighborhood Laplacian to determine the presence of edges, and thresholding with hysteresis of the gradient magnitude approximated by the Sobel operator to detect the horizontality or verticality of the edges [14].

There has been some research work on detecting edges in the compressed domain using DCT coefficients. Shen et al. [15] first reported on extracting edge information such as strength and location directly from compressed data by examining a few of the DCT AC coefficients. Li et al. [16] derived a series of rules based on linear ideal edge models to classify edges in a block of size 8x8. These rules only examine a subset of DCT AC coefficients. Lee et al. [17] uses DCT edge features comprising of horizontal and vertical DCT coefficients and thresholds them to determine the presence and orientation of edges.

In our work, we only need to determine the presence of horizontal or vertical edges, but with high confidence. Hence, we used the idea as presented by Liang et al. [16], but modified it to suit our purposes. Denoting the intensity of a 8x8 image block by $f(x,y)$, and its 2-D DCT by $F(u,v)$, where $0 \le x,y,u,v \le 7$, we have:

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\left(\frac{(2x+1)u\pi}{16}\right)\cos\left(\frac{(2y+1)v\pi}{16}\right)$$

where

$$C(\omega) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \omega = 0 \\ 1 & \text{for } \omega = 1,2,..7 \end{cases}$$

The normalized 2-D DCT, $\overline{F}(u,v)$, is defined by:

$$\overline{F}(u,v) = \frac{F(u,v)}{C(u)C(v)}$$

The horizontal, vertical and texture feature subsets are given by

$$H = \{(0,1),(0,2),(0,3),(0,4)\}$$
$$V = \{(1,0),(2,0),(3,0),(4,0)\}$$
$$T = \{(1,1),(2,2)\}$$

respectively. Hence, the subset of 2-D DCT coefficients that will be examined is given by:

$$B = H \cup V \cup T$$

An 8x8 image block will be classified as having a horizontal edge if the following conditions hold:

a) $\left|\overline{F}(0,1)\right| = \max\left(\left|\overline{F}(u,v)\right|\right) \quad \forall (u,v) \in B$

b) $\left|\overline{F}(0,1)\right| > \alpha$

c) $\left|\overline{F}(u,v)\right| < \beta\left|\overline{F}(0,1)\right| \quad \forall (u,v) \in V \cup T$

where α and β are free parameters. Here, α = 70, and β = 0.2. In a similar fashion, an 8x8 image block will be classified as having a vertical edge if:

a) $\left|\overline{F}(1,0)\right| = \max\left(\left|\overline{F}(u,v)\right|\right) \quad \forall (u,v) \in B$

b) $\left|\overline{F}(1,0)\right| > \alpha$

c) $\left|\overline{F}(u,v)\right| < \beta\left|\overline{F}(1,0)\right| \quad \forall (u,v) \in H \cup T$

The thresholding is done in recognition of the fact that in typical images, edges can be viewed as idealized edge with additive noise. Hence, α serves the purpose of only allowing significant edge differences so as to differentiate it from just random disturbances. β is set to a value above 0 because typical edges will not be perfectly horizontal or vertical, hence, some tolerance is needed. A larger α leads to more significant edges being found, while a smaller β leads to edges being found with higher confidence.

3.4. Candidate sub-windows search

An exhaustive search is done to consider all possible windows of a target size. While this operation is $O\left(N^4 \times O(\text{feature calculation})\right)$ computationally, the fact that the search is carried out in MB space together with judicious pruning makes the search more tractable.

For each considered sub-window, the following features are computed:

1) *EdgeScore*, the percentage of correct edges (i.e. vertical edge along vertical border, horizontal edge along horizontal border) along perimeter.
2) *ContentScore*, the percentage of sub-window area with MBs that are in *S(n)*, where *S(n)* is the set of MBs that are likely to be part of a sub-window shot in frame *n*. This set is updated by the equation:
$$S(n) = W(n-1) \cup L(n)$$
where *W(n-1)* is the set of a certain fraction of MBs that are part of a declared candidate sub-window shot in the previous frame.
3) *Area*, the percentage of frame area that the sub-window occupies
4) *AR*, the aspect ratio (width/height)
5) *Cdiff(I,O)*, the distance between the histograms of the inside area *I* and corresponding outside area *O* (of equal size to *I*) as shown in figure 3(a), where
$$Cdiff(I,O) = d_{L1}(H(I),H(O))$$
where *H(I)* is a 6x6 bin histogram of the area of the interest on its two color components, *Cr* and *Cb*. The bins are selected so that each bin would have approximately equal counts under the assumption that Cr and Cb were both distributed normally with mean 128 and a standard deviation of 16. $d_{L1}(H1,H2)$ is the L_1-norm between the histograms *H1* and *H2*, and is given by:
$$d_{L1}(H1,H2) = \sum_l |H1_l - H2_l|$$
A total of 8 distances are computed, corresponding to *Cdiff(UL,0)*, *Cdiff(UL,1)*, *Cdiff(UR,2)*, *Cdiff(UR,3)*, *Cdiff(LR,4)*, *Cdiff(LR,5)*, *Cdiff(LL,6)* and *Cdiff(LL,7)*. This distances will help determine if the sub-window is a distinct entity, or is a part of the surrounding background.
6) *BorderCoherence*, the maximum percentage of border MBs that are within a continuous 4x4 box within a 16x16 equal-width bin histogram of the border MB on its two color components. As shown in figure 3(b), both the outside border and inside border are considered, and the larger score will be used.

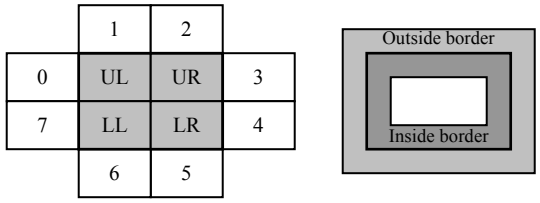


Figure 3. Areas of interest for chrominance analysis. The bold outlined box indicates the candidate sub-window under investigation. (a) Each quadrant is compared with its 2 neighboring areas. (b) Both the inside and outside borders are examined for coherence.

7) *Uniformity*, the maximum percentage of MBs that are in a bin of a 6x6 bin histogram of the candidate sub-window MBs on its two color components. The bins are similar to the ones used to calculate *Cdiff(I,O)*. A desirable sub-window should have a low *Uniformity* score.

Histograms are used to model the distribution of color components of MBs because they have been shown to have good indexing capabilities [18]. However, care has to be taken when applying it because of the small number of MBs involved. Hence, to compare difference in computing *Cdiff(I,O)*, a small histogram size is used, while to check coherence in computing *BorderCoherence*, a larger histogram size is used.

A candidate sub-window is declared if the features fall into a pre-determined sub-space. Overlaps are resolved by picking the sub-window with the best *EdgeScore* followed by *Area*.

At the same time, the number of localized MB scene change is tracked. If the fraction of local scene change is greater than δ_{GLOBAL}, then it is also considered a global scene change. In this case, it is necessary that all sub-windows be purged, and this message will be passed along to the spatial-temporal filter.

3.5. Spatial-temporal filtering

The objectives of the filter are:

- To filter out candidate sub-windows which are isolated in time, since these are probably noise.
- To obtain smoothed dimensions of the sub-window shot, since they are usually stationary in space for the duration of the shot.

A non-linear spatial temporal filter is used and implemented by the following algorithm:

1) For all candidate sub-windows identified in a frame, check to see if it overlaps with any sub-window being tracked. If there is an overlap of more than $\delta_{OVERLAP}$, go to step 3. Once all candidate sub-windows are considered, go to step 4.
2) If there is no overlap, register the sub-window in the list of sub-window shots being tracked. Repeat step 1 for other candidate sub-windows.
3) If there is an overlap, update the following parameters of the tracked sub-window:
   - Maximum temporal extent
   - Maximum spatial extent
   - Frequency of MB belonging to this candidate sub-window

   Repeat step 1 for other candidate sub-windows.
4) Check the list of sub-window shots being tracked. If there is no overlap with any candidate sub-windows from the current frame, then the tracked sub-window has ended. The best fit window is the smallest rectangular window that encompasses all the MBs that appear for more than 50% of the shot duration. Check to see if it is valid using the following rules:
   - Temporal extent is greater than time δ_{TIME}.
   - Aspect ratio of the best-fit window is valid
   - Area of the best-fit window is valid

   Then, purge this shot from the list of sub-window shots being tracked.

There are two parameters in this filter, $\delta_{OVERLAP}$ and δ_{TIME}. $\delta_{OVERLAP}$ controls how much a window can shift around; the higher its value, the more static the window has to be for it to be considered to be the one and the same window. It is set to 65%. δ_{TIME} is the shortest interval possible for a sub-window shot, and it is set to 1s for the experiment. The higher its value, the longer the window has to be around for it to be considered as a sub-window shot.

4. Experimental results and performance

To test the effectiveness of the proposed framework, the algorithm was used on four compressed MPEG sequences. The experimental setup, results and timing performance are reported in the following subsections.

4.1. Experimental setup

Four different news sequences were recorded and encoded in MPEG-1. Their descriptions are provided in Table 1.

Ground truth was obtained by viewing the news video and taking note of the temporal and spatial extent of each rectangular sub-window shot.

Table 1. Details of News Sequences

No.	Description	Number of sub-windows shots	Length (hh:mm:ss)
1.	Channel 5 News	22	00:27:37
2.	Channel U News	6	00:33:24
3.	Straits Times News	10	00:17:23
4.	CNA News	71	02:32:43
	Total:	109	03:51:07

The sequences were divided into 2 groups. Sequences 1-3 were used as training data to fine-tune the free parameters in the algorithm as described in Section 3. Sequence 4 was used as testing data to observe how well the algorithm can apply to untrained data.

4.2. Experimental results

Table 2 summarises the results of the experiment for each sequence, while table 3 groups the results for training and testing data.

Table 2. Results for individual sequences

No.	Hits	False Positives	Recall (%)	Precision (%)
1.	22(/22)	2(/24)	100	92
2.	4(/6)	6(/10)	67	40
3.	5(/10)	4(/9)	50	56
4.	16(/71)	43(/59)	23	27

Table 3. Results for individual data sets

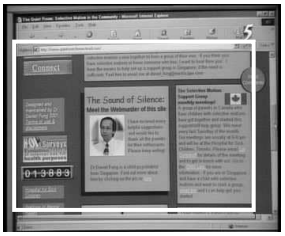
Data set	Hits	False Positives	Recall (%)	Precision (%)
Training Data	31(/38)	12(/43)	82	72
Test Data	16(/71)	43(/59)	23	27

The framework shows reasonably good performance for the training data. Unfortunately, it does not manage to do as well for the test data, which suggests that the parameters for the algorithm may have to be selected adaptively. Figure 4 gives examples of hits, false positives and misses.

(a) Straits Times News

(b) Big sub-window

(c) Advertisement with message frame



(d) Computer screen

(e) Sub-window with low contrast

(f) Two adjacent sub-windows with thin boundaries

Figure 4. Examples of hits (a-b), false positives (c-d) and misses (e-f). Notice how the algorithm always finds very distinct rectangles. Most misses are sub-windows either low contrast between itself and the surrounding, or adjacent sub-windows which are largely similar in content.

Most of the false detections capture distinct rectangular regions such as computer screens, weather reports, doors, windows, signs and logos. While some of this may arguably be sub-windows (like computer screens and weather reports), some analysis of the content can try to distinguish between natural images and synthetic images. This in turn suggests that a more precise definition of "sub-window shot" is required.

The misses usually occur due to the lack of definite visual cues such as distinct boundaries and high contrast between the sub-window and its surrounding area. This is due to the shortcomings of the edge detection which is constrained to doing its work in the compressed domain. The results should be much better if more accurate and precise detection of horizontal and vertical borders was possible.

Finally, it should be recognized that all these was done in the compressed domain, and down only to the MB level. Much information is not accessible since the video is not full frame decoded. Hence, many sub-windows which are visually obvious may not appear so in the MB level.

As mentioned earlier, parameters for the algorithm may have to be selected adaptively. This is due to the fact that each broadcaster would have used different production rules for their news video. One approach for selecting parameters adaptively is to employ a mechanism that recursively updates the parameters based on the statistics available after processing each frame. Another approach is to make use of machine learning techniques and carry out supervised learning of parameters for each set of production rules used by a broadcaster. Then, an appropriate set of parameters can be used for news video from each broadcaster.

4.3. Performance timings

Table 4 lists the runtimes for the algorithm for each sequence on a P4 2.4GHz computer with 384 MB of RAM. The machine is running Windows XP.

Table 4. Runtimes

No.	Runtime (s)	Speed-up
1.	151	11x
2.	229	9x
3.	101	10x
4.	999	9x

By working in the compressed domain, this algorithm can achieve a speed of up to 11 times real time with minimal optimisations.

5. Conclusions

Sub-window shot detection is yet another media analysis tool that can be used in the segmentation of news sequences either for video retrieval purposes or for news video story segmentation. This paper has shown a framework in which this task can be accomplished. The framework contains independent modules, each of which can be fine-tuned to improve overall performance. This paper has also described an implementation of the framework and possible improvements to it.

6. Acknowledgements

We would like to express our gratitude to Li and Sethi for providing public usage of their MDC toolkit.

We also acknowledge the use of news video produced by MediaCorp and SPH Mediaworks.

7. References

[1] M. Yeung, B.L. Yeo and B. Liu, "Extracting Story Units from Long Programs for Video Browsing and Navigation", *IEEE International Conference on Multimedia Computing and Systems*, pages 296-305, 1996.

[2] D. Zhong, H. Zhang and S.F. Chang, "Clustering Methods for Video Browsing and Annotation", *SPIE Storage and Retrieval for Still Image and Video Database IV*, volume 2670, pages 239-246, 1996.

[3] B. Yeo and B. Liu, "Rapid scene analysis on compressed video", *IEEE Transactions on Circuits and Systems for Video Technology*, volume 5, issue 6, pages 533-544, 1995.

[4] J. Feng, K. Lo and H. Mehrpour, "Scene change detection algorithm for MPEG video sequence", *International Conference on Image Processing*, volume 2, pages 821-824, 1996

[5] M. Sugano, Y. Nakajima, H. Yanagihara, and A. Yoneyama, "A fast scene change detection on MPEG coding parameter domain", *International Conference on Image Processing*, volume 1, pages 888-892, 1998

[6] K. Tse, J. Wei and S. Panchanathan, "A scene change detection algorithm for MPEG compressed video sequences", *Canadian Conference on Electrical and Computer Engineering*, volume 2, pages 827-830, 1995

[7] J. Calic and E. Izquierdo, "Towards Real-Time Shot Detection in the MPEG-Compressed Domain", *Workshop on Image Analysis for Multimedia Interactive Services*, 2001

[8] J. Calic, S. Sav, E. Izquierdo, S. Marlow, N. Murphy and N.E. O'Connor, "Temporal Video Segmentation for Real-Time Key Frame Extaction", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 3632-3635, 2002.

[9] T. Shin, J. Kim, H. Lee and J. Kim, "Hierarchical scene change detection in an MPEG-2 compressed video sequence", *IEEE International Symposium on Circuits and Systems*, volume 4, pages 253-256, 1998

[10] W.A.C. Fernando, C.N. Canagarajah and D.R. Bull, "A unified approach to scene change detection in uncompressed and compressed video", *IEEE Transactions on Consumer Electronics*, volume 46, issue 3, pages 769-779, 2000

[11] D. Le Gall, "A video compression standard for multimedia applications", *Communications of the ACM*, volume 34, number 4, pages 46-58, 1991.

[12] ISO/IEC, "Information Technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s", *ISO/IEC 11172-1/2*, 1993

[13] D. Li and I.K. Sethi, "MDC: a software tool for developing MPEG applications", *IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 445-450, 1999.

[14] M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis and Machine Vision*, PWS Publishing, USA, 1999.

[15] B. Shen and I.K. Sethi, "Direct feature extraction from compressed images", *Proceedings SPIE Storage & Retrieval for Image and Video Databases IV*, volume 2670, pages 33-49, 1996.

[16] H. Li, G. Liu and Y. Li, "An effective approach to edge classification from DCT domain", *IEEE International Conference on Image Processing*, volume 1, pages 940-943, 2002.

[17] M. Lee, S. Nepal and U. Srinivasan, "Role of Edge Detection in Video Semantics", *ACS Conferences in Research and Practice in Information Technology*, volume 22, pages 59-68, 2003.

[18] M. Stricker and M. Swain, "The capacity and the sensitivity of color histogram indexing", Technical Report 94-05, University of Chicago, Mar. 1994.

[19] W. Hsu and S. Chang, "A Statistical Framework for Fusing Mid-level Perceptual Features in News Story Segmentation", *IEEE International Conference on Multimedia and Expo*, volume 2, pages 413-416, 2003.

[20] S. Chang, "The Holy Grail of Content-Based Media Analysis", *IEEE Multimedia Magazine*, volume 9, issue 2, pages 6-10, 2002.