

AUTOMATIC VIDEO CUT DETECTION USING ADAPTIVE THRESHOLDS

Oscar D. Robles

Dpto. de Informática, Estadística y Telemática.
U. Rey Juan Carlos. C. Tulipán, s/n.
28933 Móstoles. Madrid. Spain.
orobles@escet.urjc.es

Angel Rodríguez

Dept. de Tecnología Fotónica.
U. Politécnica de Madrid.
Campus de Montegancedo s/n.
28660 Boadilla del Monte. Madrid. Spain.
arodri@dtf.fi.upm.es

Pablo Toharia

Dpto. de Informática, Estadística y Telemática.
U. Rey Juan Carlos. C. Tulipán, s/n.
28933 Móstoles. Madrid. Spain.
ptoharia@escet.urjc.es

Luis Pastor

Dpto. de Informática, Estadística y Telemática.
U. Rey Juan Carlos. C. Tulipán, s/n.
28933 Móstoles. Madrid. Spain.
lpastor@escet.urjc.es

ABSTRACT

This paper presents a new automatic shot detection technique based on multiresolution histograms. Starting from an adaptive threshold filter, we extract shot cuts when multiresolution histogram peaks are significant enough. In the results section we present the recall and precision achieved on a video database composed by 61 AVI videos and 816 shots.

KEY WORDS

CBIR primitives, Video Retrieval, Video Indexing

1 Introduction

One of the main objectives of Content-based Multimedia Retrieval systems is the automatization of the information extraction process from the raw data. When dealing with video data, the first step is to perform a temporal video segmentation in order to make a shot decomposition of the video content. Del Bimbo [1], Brunelli *et al.* [2] and Hanjalic [3] collect extensive reviews of this set of techniques. Depending on the domain of work, these techniques can be classified in non-compressed [4, 5, 6, 7] and compressed video shot segmentation [8, 9, 10].

This work focuses on an analysis of the response of a new shot detection technique that filters flash effects applied over general thematic videos. The main feature of the technique herein described is the high adaptability of the algorithm to a wide range of videos due to the variable threshold managed in the shot extraction algorithm, since it has been noticed that global fixed video thresholds applied over diverse thematic videos, as it has been proposed by other authors, does not provide the expected results in all cases [11]. Several color-based features have been implemented in order to make an exhaustive analysis of the system response.

The content of this paper may be broken down into a description of the global strategy and the implementation analysis of the proposed shot extraction technique (Section

2), followed by the results achieved during the tests (Section 3) and the conclusions obtained (Section 4).

2 Shot extraction description

2.1 Global strategy

Video cut detection has two main purposes: to delimit the start and the end of the video shots and to process the video content in a more efficient way. The basic idea of video cut detection algorithms is to compute the differences between consecutive frames or groups of frames. Existing techniques differ in the way these differences are computed.

Figure 1 depicts a scheme of the whole process. D_i denotes the difference between the considered frame and the previous one. In our case, the computed D_i difference values are based on several color features in order to make a more exhaustive analysis of the system response. The features implemented have been mean intensity, histograms and multiresolution histograms [12]. A more detailed description of the implemented features can be found farther on. A candidate for cut is detected when the values are higher than a dynamically computed threshold Th . The expression of Th is defined by Eq. 1

$$Th = \text{weight} \frac{\sum_{i=j-W}^{i+W} D(i)}{2W + 1} \quad (1)$$

where W is the number of difference values of the left and right local neighbour windows, i is the frame under consideration and weight is a gain factor. Therefore, the threshold is updated for each processed frame.

One of the typical artifacts present in videos is the appearance of flashes that distort the normal analysis of the video signal, because there is no change in the video content but abrupt changes appear in signal intensity. In order to filter out the flashes, a second threshold T_{flash} has been implemented, following the model of Zhang *et al.* [11]. Finally, once the comparisons are performed the threshold

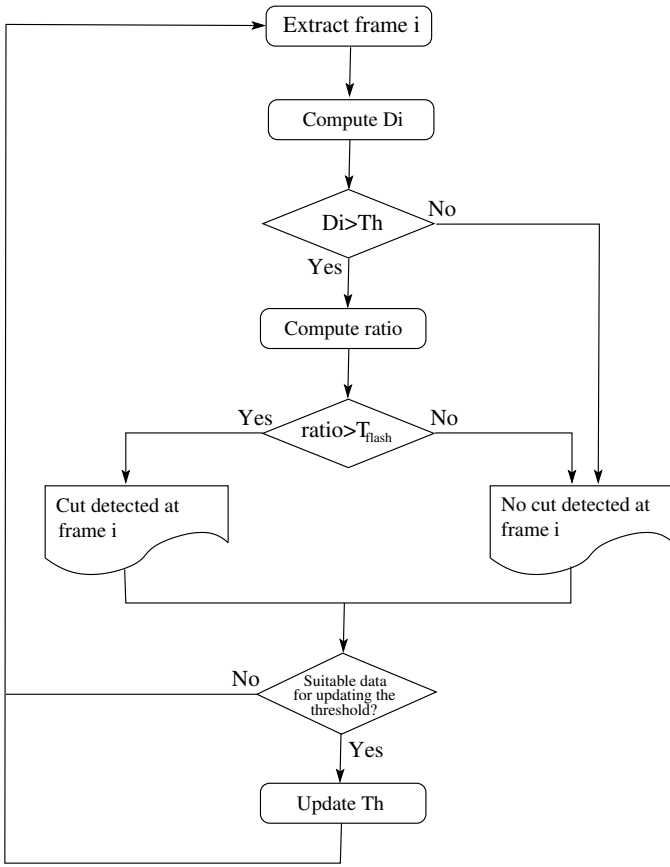


Figure 1. Cut detection algorithm.

Th is recalculated, so that value can be adapted to the new video signal content.

2.2 Implementation analysis

As Figure 1 shows, the cut detection algorithm starts extracting frame i and computing difference D_i which is compared against current threshold Th . In case this difference is greater than Th a ratio for detecting flash effects is calculated. If this ratio is greater than threshold T_{flash} a flash is detected. Otherwise, a cut is found. Finally, the current window variance is calculated in order to test whether the data is suitable for updating threshold Th , and in that case a recalculation is needed. These ideas will be explained in depth below.

Sometimes it happens that the processed difference values D_i vary too much from frame i to next frame $i + 1$, as it is the case when, for example, very fast camera movements occur. Therefore, those values that are far away from the sequence recently processed must be discarded. The criteria used to filter these outliers is the variance computed over the sliding window [11]. When the variance V_i is greater than an heuristic predefined threshold T_v , the current threshold Th is not updated.

Flash elimination is done taking into consideration

that the appearance of a flash produces an abrupt change in intensity, but unlike real cut edges, the level of the signal comes back to the previous state after one more frame or after a very few ones. The expression that filter flashes is

$$\text{ratio}_{\text{flash}} = \frac{D_s}{D_i}, \quad (2)$$

where D_s is the difference between the W frames preceding the current frame and the W ones after it.

Several color primitives have been tested: mean intensity [11], histogram [11], multiresolution histograms [12] and multiresolution energy [13]. The value $\text{ratio}_{\text{flash}}$ has been normalized for each primitive in order to work in the interval $[0, 1]$, ideally meaning respectively flash and cut:

$$\begin{cases} \text{ratio}_{\text{flash}} < T_{\text{flash}} & \text{Flash detection} \\ \text{ratio}_{\text{flash}} \geq T_{\text{flash}} & \text{Cut detection} \end{cases} \quad (3)$$

Each primitive defines different expressions for D_s :

- Mean intensity (MI):

$$D_s = \frac{1}{255W} \left| \sum_{k=i-W}^{i-1} MI_k - \sum_{k=i+1}^{i+W} MI_k \right| \quad (4)$$

- Multiresolution energy:

$$D_s = \frac{1}{N_E} \sum_{c=0}^{N_E-1} \frac{|E_{\text{left}}(c) - E_{\text{right}}(c)|}{\max(|E_{\text{left}}(c)|, |E_{\text{right}}(c)|)} \quad (5)$$

where

$$E_{\text{left}}(c) = \frac{1}{W} \sum_{k=i-W}^{i-1} E_k(c) \quad (6)$$

$$E_{\text{right}}(c) = \frac{1}{W} \sum_{k=i+1}^{i+W} E_k(c) \quad (7)$$

$$\forall c = 0, \dots, N_E$$

and N_E is the number of elements of the energy vector (see [13]).

- Histogram-based primitives:

$$D_s = \frac{1}{2} \sum_{c=0}^{255} |H_{\text{left}}(c) - H_{\text{right}}(c)| \quad (8)$$

where

$$H_{\text{left}}(c) = \frac{1}{W} \sum_{k=i-W}^{i-1} H_k(c) \quad (9)$$

$$H_{\text{right}}(c) = \frac{1}{W} \sum_{k=i+1}^{i+W} H_k(c) \quad (10)$$

$$\forall c = 0, \dots, 255$$

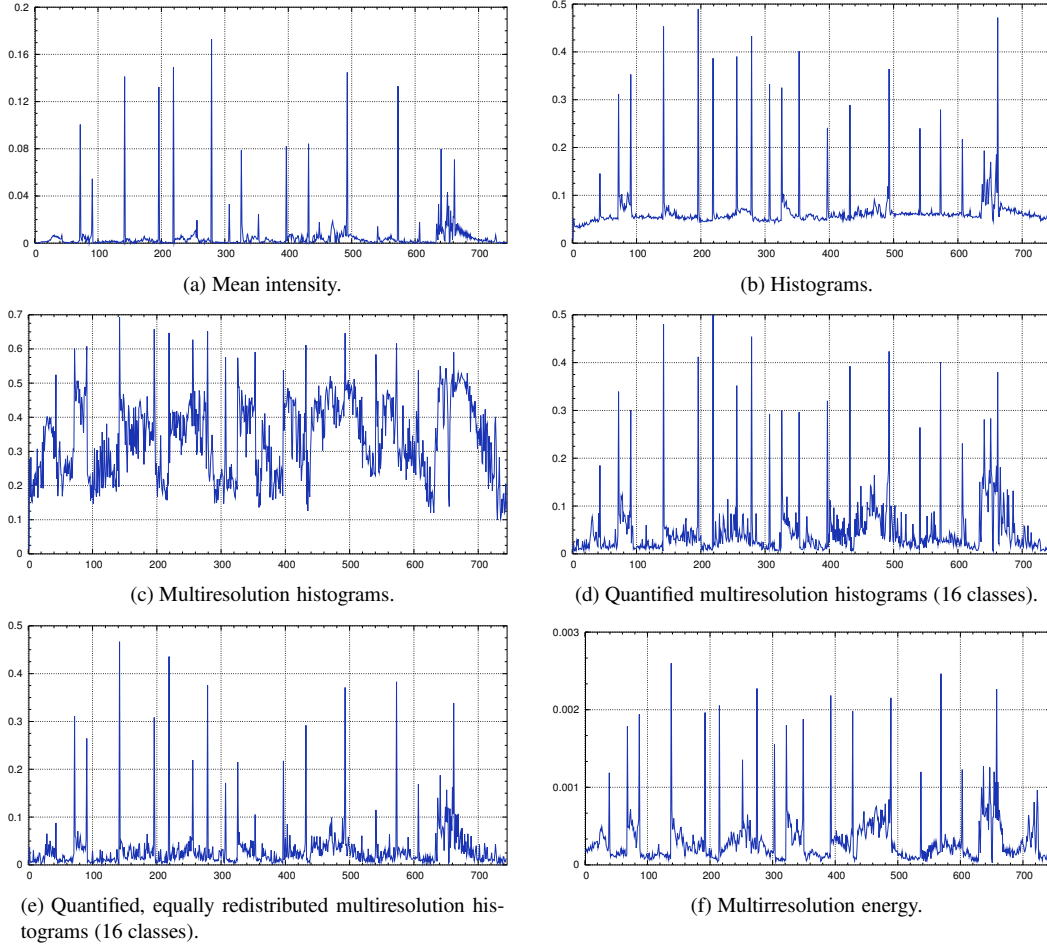


Figure 2. Comparison of graphs showing number of frames against D_i values.

We have equally subdivided the interval $[0, 1]$ to assign $D_s < 0.5$ to flashes.

Histograms usually present a problem when comparing distributions concentrated around near but not exactly equal values. The result of the comparison produces high difference values although the appearance of the images that generate the histograms is very similar. To reduce this effect a histogram quantification has been performed, grouping color values to generate new classes:

$$H_q(i) = \sum_{j=iT_q}^{iT_q+T_q-1} H(j), \forall i \in [0, N_q - 1] \quad (11)$$

where H_q is the quantified histogram, H is the original histogram, T_q is the size of the new classes and N_q is the number of classes ($N_q \cdot T_q = 256$ for 8 bit cases).

While working with the quantification, the same problem can appear around the boundaries of the classes. It can be solved equally redistributing edge levels between neigh-

bour classes. It can be achieved computing the value c_i :

$$c_i = \frac{\sum_{j=iT_q-\varepsilon}^{iT_q+\varepsilon} H(j)}{2\varepsilon} \quad (12)$$

so the frequency of class i on the resultant quantified histogram with equal redistribution $H_q^e(i)$ will be:

$$H_q^e(i) = c_i + c_{i+1} + \sum_{j=iT_q+\varepsilon}^{iT_q+T_q-1-\varepsilon} H(j) \quad (13)$$

where the summatory comes from equation 11.

Figure 2 shows a comparison among the distributions of D_i values computed over one of the videos used in the experiments by the features previously described. Each graph presents the number of frames against D_i values. It can be seen that around the frames 260 and 300, the Figure 2(a) does not show some peaks that appear in Figure 2(b) and are expected to be cuts. On other side, Figure 2(c) shows unreal differences, while Figure 2(d) shows a smoother graph.

Table 1. Precision and recall mean values achieved over the whole set of videos with fixed thresholds.

Threshold	Precision	Recall
0.2	0.748	0.659
0.4	0.669	0.341
0.5	0.224	0.553

Once the cuts have been detected, a formalization of the edges extraction is written to an XML description [14] of the extracted shots. In order to summarize the content of a shot, we have chosen a key frame per shot, specifying the beginning, the end, and the key frame for every detected shot.

3 Experimental Results

3.1 Experiments Setup

The main objectives of the tests are to measure and analyze the recall and precision values of the implemented features with and without the dynamic threshold. We use the classical definition of recall and precision:

$$\text{Recall:} = \frac{\text{True positives}}{\text{True positives} + \text{True negatives}} \quad (14)$$

$$\text{Precision:} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \quad (15)$$

We have created a video database composed by 61 general thematic AVI videos that have been manually segmented on 816 shots, producing 755 cuts. Finally, we have computed the response of each one of the implemented features and threshold combinations over the whole database.

All the tools involved in the software developed are free distribution tools, like vs. 2.4.18-14 Linux operating system, vs. 3.2.7 of the GCC GNU compiler [15], vs. 0.7.34 of the AVI file processing library AVIFILE [16] and vs. 2.4.23 of the LIBXML2 library for processing XML files [17].

3.2 Results analysis

Tables 1 and 2 show the recall and precision mean values computed taking into consideration the 61 videos.

Table 1 contains a summary of the precision and recall values for all the primitives herein described but using fixed thresholds. Although some fixed thresholds can achieve good results on specific videos using certain primitives, the balance between the average values of precision and recall, as can be noticed on Table 1, is unsatisfactory, justifying the need to improve this approach.

The suffix notation used in the method column in Table 2 is the following: MI —mean intensity—, EN —energy—, H —histogram—, Q —quantified—, MR —multiresolution—, E —equally redistributed— and number

Table 2. Precision and recall values achieved over the whole set of videos with adaptive threshold.

Method	Precision	Recall
dif_MI	0.399	0.813
dif_ENMR	0.820	0.590
dif_H	0.892	0.582
dif_H_Q4	0.820	0.862
dif_H_Q8	0.866	0.867
dif_H_Q16	0.903	0.856
dif_H_Q32	0.917	0.837
dif_H_QE4	0.656	0.844
dif_H_QE8	0.802	0.860
dif_H_QE16	0.850	0.858
dif_H_QE32	0.876	0.854
dif_HMR	0.400	0.130
dif_HMR_Q4	0.436	0.847
dif_HMR_Q8	0.537	0.839
dif_HMR_Q16	0.641	0.773
dif_HMR_Q32	0.768	0.706
dif_HMR_QE4	0.348	0.825
dif_HMR_QE8	0.466	0.831
dif_HMR_QE16	0.594	0.814
dif_HMR_QE32	0.665	0.787

Table 3. Mean computation time per video and per frame, measured in seconds.

Method	Mean time	Mean time / frame
dif_IM	3,64321	0,00462
dif_ENMR	5,00747	0,00621
dif_H	3,63338	0,00460
dif_H_C4	3,63130	0,00460
dif_H_C8	3,66899	0,00464
dif_H_C16	3,66497	0,00463
dif_H_C32	3,62029	0,00458
dif_H_CM4	3,62435	0,00460
dif_H_CM8	3,64432	0,00461
dif_H_CM16	3,66173	0,00463
dif_H_CM32	3,66859	0,00465
dif_HMR	4,85157	0,00611
dif_HMR_C4	4,87117	0,00611
dif_HMR_C8	4,86818	0,00612
dif_HMR_C16	4,93474	0,00617
dif_HMR_C32	4,85560	0,00611
dif_HMR_CM4	5,03465	0,00626
dif_HMR_CM8	5,08669	0,00628
dif_HMR_CM16	5,08562	0,00630
dif_HMR_CM32	5,03422	0,00626

—number of classes—. We have tested several values for the quantified histograms. Analyzing the Table 2, it can be seen that in most cases, with the dynamic computed threshold, the recall and precision values are very high

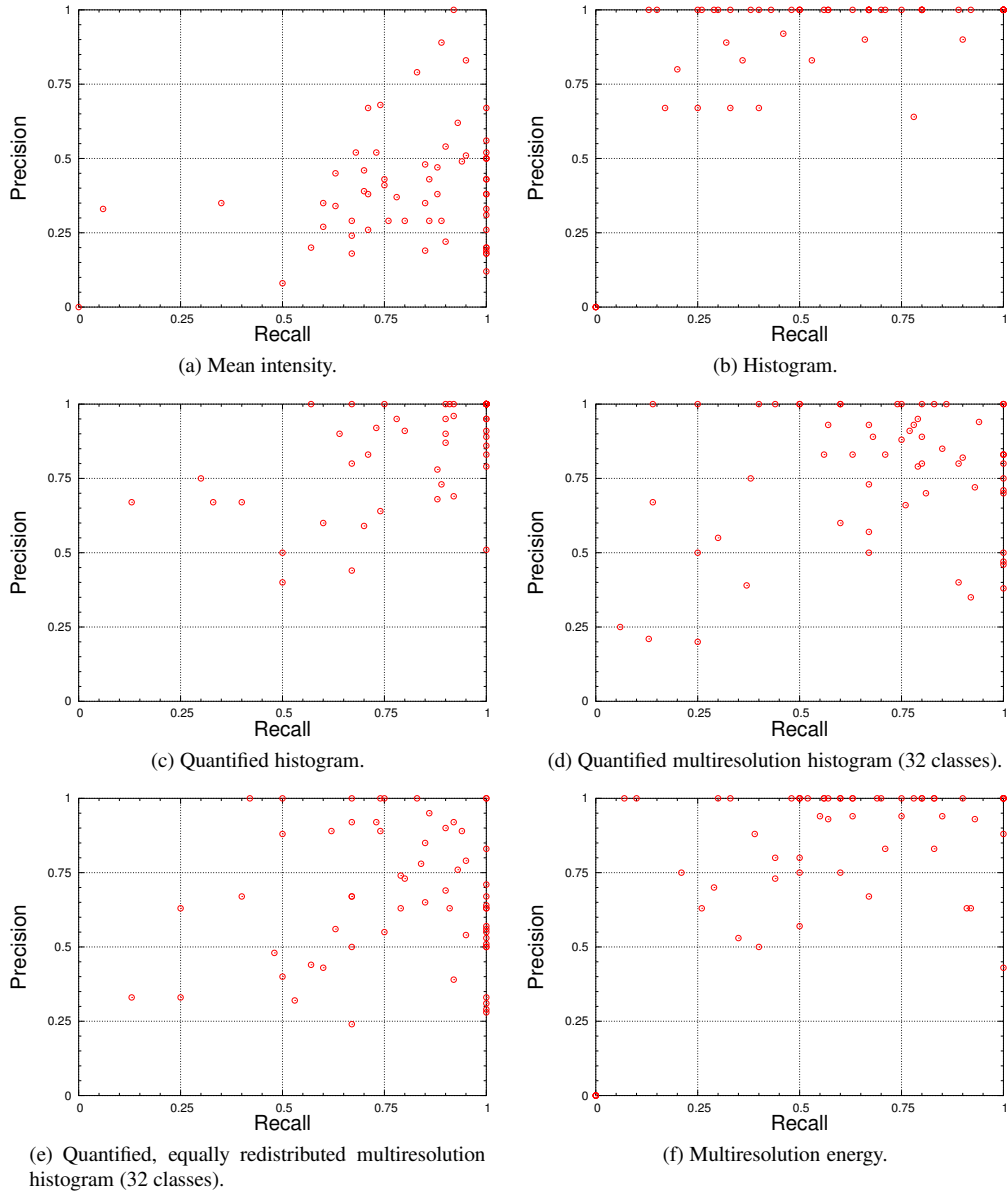


Figure 3. Recall-precision graphic measures at shot level.

keeping a good balance between them, while the results achieved with fixed thresholds are very disappointing. As can be noticed in the table, quantified histograms improve the results obtained by histograms and multiresolution histograms. The feature that obtains the best recall value is the one that uses quantified histograms with 8 classes. On the other hand, precision values are not so high as recall measures, but in most cases exceed 50%. It is remarkable the values achieved by quantified histograms with 32 classes, above 90%. Although multiresolution histograms have obtained better results than histograms in CBIR applications [12], in this case, low displacements on multiresolution histograms reduce significantly the matching measure for shot edge detection.

Figure 3 shows the combination of recall and preci-

sion values for the implemented features at shot level. The Figures show the best distributions for mean intensity 3(a), histogram 3(b), quantified histogram 3(c), quantified multiresolution histogram with equal redistribution 3(d) and multiresolution energy. The dispersion shown by the mean intensity graph 3(a) is highly reduced by the histograms. The use of quantized histograms with equal redistribution, Figures 3(c) and 3(d), improves the behavior of the multiresolution histograms. The best results are obtained for quantified histograms with 16 and 32 classes.

In order to evaluate the computational cost of this methods, their execution times have been measured. The measures have been taken using a PC Pentium 4 with 3 GHz of clock frequency, hyperthreading and 1 GB of RAM memory. The data thus obtained is resumed on Table 3.

Due to the variable properties of the videos used, it has also been computed the mean processing time per frame. This table gives a good idea of the methods efficiency although it should be taken into account that there is no normalization in the frame size.

4 Conclusions and future work

In this paper, a new strategy for video cut detection is presented. High recall and precision values are achieved using adaptive dynamic thresholding. The primitives have been tested on a video database which contains 61 general thematic AVI videos with 755 manually segmented shots. Between the implemented features, quantified histograms provide the best results.

Future work will be focused on a more efficient implementation of the threshold Th updating. Alternatives to variance computation will be studied. Apart from that, a combination of the primitives may improve the precision and recall measures. Gradual transitions between shots will also be considered in the future.

Acknowledgments

This work has been partially funded by the Spanish Commission for Science and Technology (grant CICYT TIC2003-08933-C02-01).

References

- [1] Alberto del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, California, 1999. ISBN 1-55860-624-6.
- [2] R. Brunelli, O. Mich, and C. M. Modena. A survey on video indexing. *Journal of Visual Communication and Image Representation*, 10:78–112, 1999.
- [3] Alan Hanjalic. Shot-boundary detection: Unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):90–105, February 2002.
- [4] Sara Porter, Majid Mirmehdi, and Barry Thomas. Temporal video segmentation and classification of edit effects. *Image and Vision Computing*, 21(13–14):1097–1106, December 2003.
- [5] Min Gyo Chung, Hyeokman Kim, and S. Moon-Ho Song. A scene boundary detection method. In *Proceedings of the International Conference on Image Processing 2000, ICIP 00*, volume 3, pages 933–936, Vancouver, September 2000. IEEE Computer Society.
- [6] G. Valencia, J. A. Rodríguez, C. Urdiales, and F. Sandoval. Color-based video segmentation using interlinked irregular pyramids. *Pattern Recognition*, 37(2):377–380, February 2004.
- [7] Rozenn Dahyot, Niall Rea, and Anil Kokaram. Sport video shot segmentation and classification. In Tonradj Ebrahimi and Thomas Sikora, editors, *Visual Communications and Image Processing 2003*, volume 5150, pages 404–413, Univ. of Italian Switzerland (USI), Lugano, Switzerland, July 2003. SPIE. ISBN 0-8194-5023-5.
- [8] Sameer Antani, Rangachar Kasturi, and Ramesh Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35(4):945–965, April 2002.
- [9] Robert A. Joyce and Bede Liu. Temporal segmentation of video using frame and histogram-space. In *Proceedings of the International Conference on Image Processing 2000, ICIP 00*, volume 3, pages 941–944, Vancouver, September 2000. IEEE Computer Society.
- [10] Hongjiang Zhang. Video content analysis and retrieval. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, chapter 5.5, pages 945–977. World Scientific Publishing Company, 1998.
- [11] Dong Zhang, Wei Qi, and Hong Jiang Zhang. A new shot boundary detection algorithm. In Heung-Yeung Shum, Mark Liao, and Shih-Fu Chang, editors, *IEEE Pacific Rim Conference on Multimedia*, volume 2195, pages 63–70. IEEE, Springer, October 2001.
- [12] Oscar D. Robles, Angel Rodríguez, and M. Luisa Córdoba. A study about multiresolution primitives for content-based image retrieval using wavelets. In M. H. Hamza, editor, *IASTED International Conference On Visualization, Imaging, and Image Processing (VIIP 2001)*, pages 506–511, Marbella, Spain, September 2001. IASTED, ACTA Press. ISBN 0-88986-309-1.
- [13] Angel Rodríguez, Oscar D. Robles, and Luis Pastor. New features for Content-Based Image Retrieval using wavelets. In Fernando Muge, Rogério Caldas Pinto, and Moisés Piedade, editors, *V Ibero-american Symposium on Pattern Recognition, SIARP 2000*, pages 517–528, Lisbon, Portugal, September 2000. ISBN 972-97711-1-1.
- [14] Extensible Markup Language (XML) 1.0 (Second Edition). Web, oct 2000. W3C Recommendation. <http://www.w3.org/TR/REC-xml>
- [15] GNU. <http://www.gnu.org>
- [16] Linux avifile library. Web. <http://avifile.sourceforge.net>
- [17] Gnome Project. Gnome XML C parser and toolkit. Web. <http://www.xmlsoft.org>