COMBINING SHAPE AND COLOR FOR AUTOMATIC VIDEO CUT DETECTION

Pablo Toharia [1], Oscar D. Robles[1], Ángel Rodríguez[2] and Luis Pastor[1]

[1]Dpto. de Informatica, Estadistica y Telemática.
U. Rey Juan Carlos. C/ Tulipán, s/n. 28933 Móstoles. Madrid. Spain.
{pablo.toharia,oscardavid.robles,luis.pastor}@urjc.es

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

ABSTRACT

This paper presents some automatic shot segmentation techniques based on shape features and an analysis of their combination with color one. In the case of shape, Zernike invariants have been chosen because of its good performance for object recognition. In the case of color, the previous experience during TRECVID04 using standard and multiresolution color histograms has been taken into account. Several alternatives for combining both features for cut video detection have been tested with very positive results in some cases. The approaches tested in the submitted runs are mainly focused on cut detection. The main goal of the runs is to study the behaviour of the shape primitive individually and, appart from that, how their combination with a color-based approach improves the results obtained.

It can be noticed that the techniques herein described improve in some cases the results presented at the TRECVID04, showing a path to follow to turn them into a powerful tool to work on shot segmentation.

KEY WORDS
Shot Segmentation, CBIR primitives, Video Retrieval, Video Indexing

1 Introduction

One of the main objectives of Content-based Multimedia Retrieval systems is the automation 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 the study of the behaviour of shape features for cut detection using adaptive thresholds. Starting from the Zhang *et al.* description of the technique [11], and taking into account that it did not provide the expected results in all cases, a new implementation had been made, introducing improvements oriented to work with multiresolution histograms [12]. The work herein described presents an adaptation of that new implementation for working with shape features based on Zernike invariants, and also its combination with information extracted from color features. The main feature of the technique herein described is its high adaptability to a wide range of videos due to the variable threshold managed.

The contents of this paper may be broken down into a description of the proposed shape based shot extraction technique (Section 2), continuing with a description of the way color and shape information is mixed (Section 4), then the implementation analysis (Section 3), followed by the results achieved during the tests (Section 5) and the conclusions obtained (Section 6).

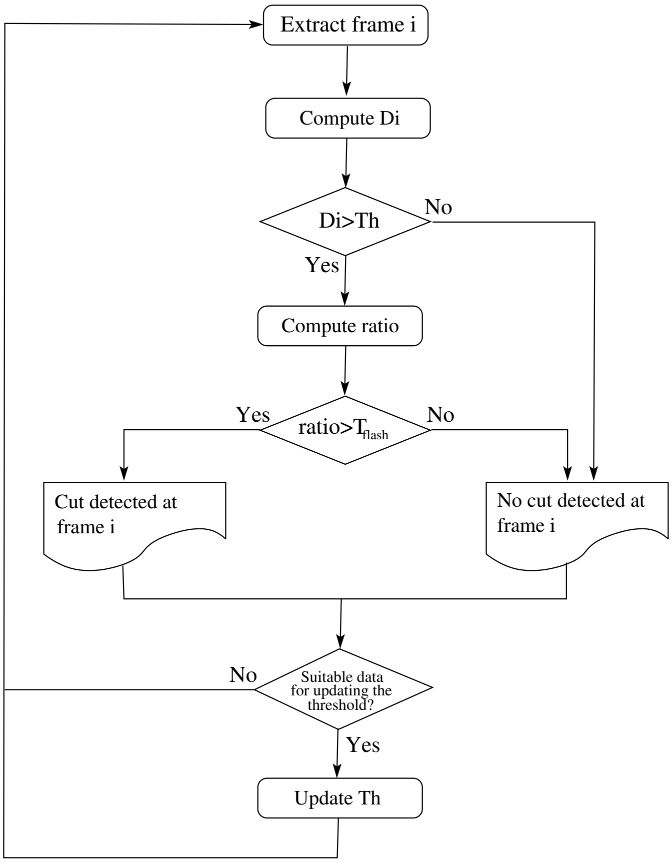


Figure 1: Cut detection algorithm.

2 Cut video detection

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. As stated in Section 1 it is the unavoidable first step to proceed with new data in a Content-based Multimedia Retrieval process. But it is also a very interesting test bench for studying more deeply and improving techniques coming from Content-based Image Retrieval in a straightforward way [13].

In the following, the global approach followed for video cut extraction will be described and then, the selected shape primitive used for this purpose will be introduced.

2.1 Global strategy

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 this case, the computed D_i difference values are based on several shape and color features in order to make a more exhaustive analysis of the system response.

The features implemented have been Zernike invariants for the shape primitive and quantified histograms for the color feature [14]. A more detailed description of the implemented shape features can be found further 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

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

where W is the number of difference values taken into account 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 Th is recalculated, so that value can be adapted to the new video signal content.

As Figure 1 shows, the cut detection algorithm starts extracting the frame i and computing the difference D_i which is compared against current threshold Th. If this difference is greater than Th a ratio for detecting flash effects is calculated. When 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 the value of threshold Th, and in that case a recalculation is needed.

2.2 Zernike invariants

The objective is to define a primitive that collects the shape information of an image in order to refine the video cut extraction based only on color [12]. Zernike invariants have been selected because of its demonstrated good performance in object recognition problems [15, 16].

In 1934, Zernike [17] presented a set of complex polynomials $V_{nm}(x, y)$ that were defined inside a unity radius circle $(x^2 + y^2 \leq 1)$ in the following way:

$$V_{nm}(x,y) = V_{nm}(\rho,\theta) = R_{nm}(\rho)\ e^{jm\theta} \tag{2}$$

where V_{nm} is a complete set of complex polynomials, n is a positive integer value $n \geq 0$ that represents the polynomial degree, m is the angular dependency and must complain that

$$(n - |m|) \bmod 2 = 0 \quad \text{and} \quad |m| \leq n \tag{3}$$

ρ and θ are the polar coordinates of the Cartesian coordinates (x, y) and R_{nm} is a set of radial polynomials that have the property of being orthogonal inside the unity circumference. These functions have the following expression:

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{m-|n|}{2}} (-1)^s \frac{(m-s)!}{s!(\frac{m+|n|}{2}-s)!(\frac{m-|n|}{2}-s)!}\rho^{m-2s} \tag{4}$$

It must be remarked that $R_{nm} = R_{n\,-m}$ and also that not satisfying the condition expressed in Eq. 3 makes $R_{nm}(\rho) = 0$.

Starting from Zernike polynomials and projecting the function over the orthogonal basis composed by the polynomials, the moments can be generated in the following way:

$$A_{mn} = \frac{m+1}{\pi}\iint_{x^2+y^2\leq 1} f(x,y)V_{nm}^*(x,y)dxdy \quad \text{with} \quad x^2+y^2 \leq 1 \tag{5}$$

The discretization needed to work with digital data can be done straightforwardly:

$$A_{mn} = \frac{m+1}{\pi}\sum_x\sum_y f(x,y)V_{nm}^*(x,y)dxdy \quad \text{with} \quad x^2+y^2 \leq 1 \tag{6}$$

From these functions, we compute the modulus to obtain the p different invariant values for each considered case. The invariant values are used to create a vector of p elements ZI_i that collect the shape information of a frame i. For example, in the case of polynomials up to tenth degree, p would be 36. These vectors are used to obtain the value D_i that determines if two consecutive frames are different enough to be considered as a shot boundary:

$$D_i = \text{dist}(ZI_i, ZI_{i-1}) \tag{7}$$

where dist refers to the Euclidean distance.

3 Implementation analysis of shape based feature

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

$$\text{ratio}_{\text{flash}} = \frac{D_{\text{s}}}{D_i}, \tag{8}$$

where D_{s} is the difference between the W frames preceding the current frame and the W ones after it and D_{i} has been defined in Sec. 2.1. The value $\text{ratio}_{\text{flash}}$ has been normalized in order to work in the interval $[0, 1]$, ideally meaning flash ($ratio_{flash} = 0$) and cut ($ratio_{flash} = 1$):

$$\begin{cases} \text{ratio}_{\text{flash}} & < & T_{\text{flash}} & \text{Flash detection} \\ \text{ratio}_{\text{flash}} & \geq & T_{\text{flash}} & \text{Cut detection} \end{cases} \tag{9}$$

For the chosen Zernike moment based invariants, the expression D_{s} is:

$$D_{\text{s}} = \text{dist}\left(\frac{1}{W}\sum_{k=i-W}^{i-1} ZI_k, \frac{1}{W}\sum_{k=i+1}^{i+W} ZI_k\right) \tag{10}$$

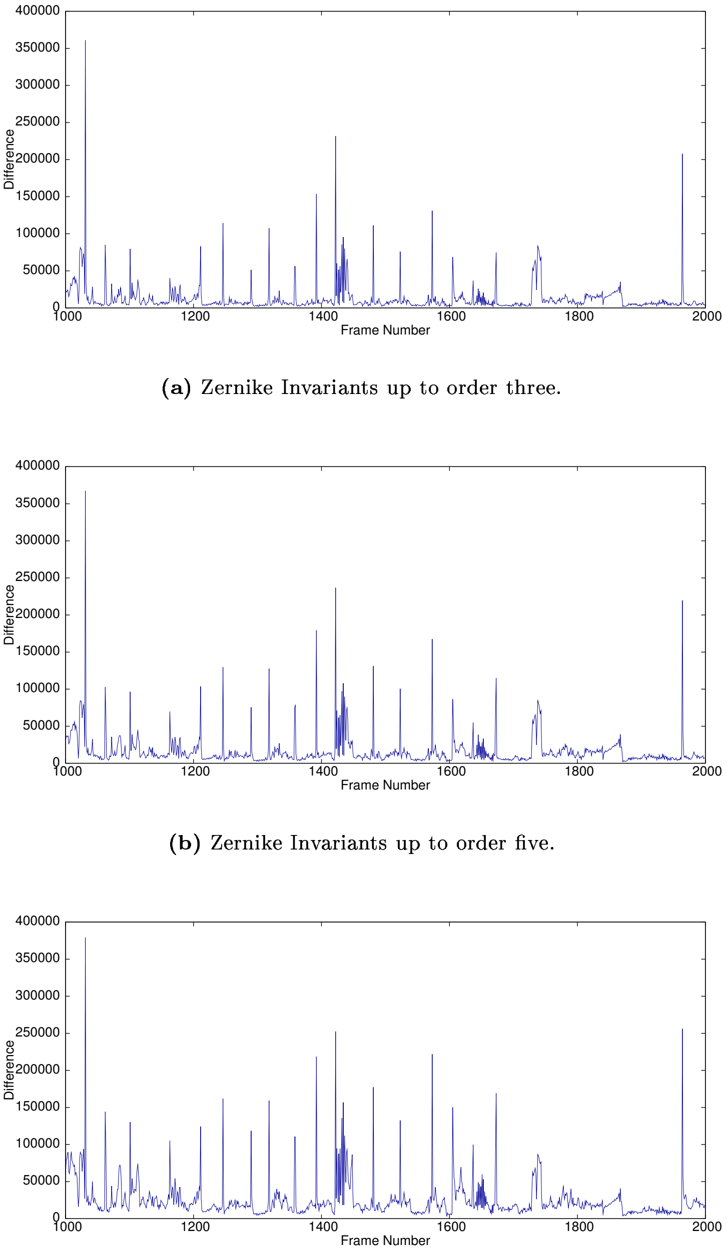
being dist the Euclidean distance.

The interval $[0, 1]$ has been equally subdivided to assign $D_{\text{s}} < 0.5$ to flashes.

As it can be deduced from Eqs. 4 and 6, the computation of the invariants is a very high demanding task from a computational point of view, so polynomials of different orders have been tested in order to verify if there were significant differences between their responses. Figure 2 shows the values D_i obtained considering several orders for the Zernike invariants.

Another issue faced in the implementation is the mapping of the rectangular domain of the processed frames to the circular space where the radial polynomials are defined (Eq. 2). The radius unity circle has been inscribed into the frame, so its corners have been discarded under the assumption that they do not usually contain relevant information about the scene (Fig. 3).

Once the cuts have been detected, the boundaries of the shots extracted are written to a file using an XML formal description. 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.



(a) Zernike Invariants up to order three.

(b) Zernike Invariants up to order five.

(c) Zernike Invariants up to order ten.

Figure 2: Comparison of graphs showing D_i values obtained by several orders Zernike Invariants.

4 Mixing shape and color

The selection of a color primitive for its combination with the presented shape primitive has been influenced by the previous experience on shot segmentation using standard color histograms, as well as global multiresolution histograms computed over the analysis coefficients of the frame's wavelet transform. Actually, the best results reported during TRECVID04 were obtained quantifying standard histograms to 16 bins [14].

In order to fuse the information proceeding from both primitives, shape and color, three approaches have been followed considering several levels of demand about what is considered a cut:

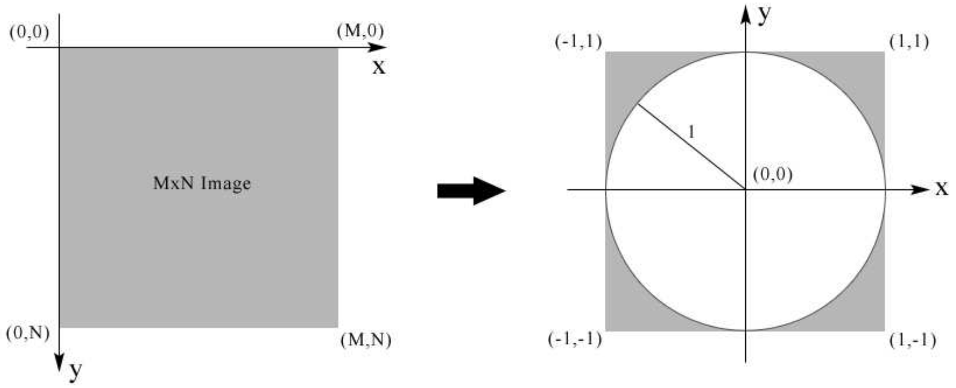


Figure 3: Inscribing Zernike's polynomial definition domain into rectangular frames.

1. Consider the set of simultaneously detected cuts both on shape and color primitives (AND).
2. Consider the sum of independently detected cuts on shape or color primitives (OR).
3. Same case as 2 but introducing a certainty value *UM*, ranged in the interval [0,1].

   The certainty value UM is only computed when a cut has been detected, *i.e.* $D_s > Th$. It is obtained from the expression

   $$UM = 1 - \frac{Th}{D_s} \tag{11}$$

   so a greater difference between the threshold Th and the current difference value D_s implies that the certainty value UM is higher. The certainty value can be used to filter those cuts detected on both primitives having low levels of confidence. In this case, since both certainty values for shape and color are collected, the sum takes values into the interval [0,2], and those cuts are discarded when

   $$UM = UM_s + UM_c \leq 0'5 \tag{12}$$

   being UM_s and UM_c respectively the color and shape certainty values. The fixed threshold can be dynamically updated considering some statistical variable, although there are satisfactory results with this value.

5 Experimental Results

5.1 Experiments Setup

The main objectives of the tests are to measure and analyze the recall and precision values of the implemented shape feature and its combination with color information. The classical definition of recall and precision has been used:

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \tag{13}$$

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \tag{14}$$

The results shown in the following sections correspond to five runs evaluated by the TRECVID05 team and one with our own evaluation using the available official software test.

The approaches tested in the submitted runs are:

- ZER3_1: Differences of Zernike moment invariants up to third order polynomials.

Method	Recall	Precision
ZER3_1	0.964	0.346
ZER5_1	0.953	0.370
ZER10_1	0.922	0.411
H_C16	0.922	0.773
ZER10_O_HC16_1	0.967	0.408
ZER10_O_HC16_UM_1	0.778	0.902
ZERO10_A_HC16_1	0.820	0.886

Table 1: Precision and recall obtained for cut detection evaluated by TRECVID05.

Method	Total decode time	Total segmentation time
ZER3_1	3504	52452
ZER5_1	3540	100123
ZER10_1	3611	350609
H_C16	3583	1492
ZER10_O_HC16_1	(3611, 3583)	(350609, 1492)
ZER10_O_HC16_UM_1	(3611, 3583)	(350609, 1492)
ZERO10_A_HC16_1	(3611, 3583)	(350609, 1492)

Table 2: Processing complexity.

- ZER5_1: Differences of Zernike moment invariants up to fifth order polynomials.
- ZER10_1: Differences of Zernike moment invariants up to tenth order polynomials.
- ZER10_O_HC16_1: OR Combination of color and shape using differences of Zernike moment invariants up to tenth order polynomials and differences of color histograms, quantified to 16 classes or bins, with redistribution of boundary values.
- ZER10_O_HC16_UM_1: Idem as ZER10_O_HC16_1 but introducing the certainty value *UM*.

The following cases not previously submitted have been also tested, using the tools provided by the TRECVID05 team:

- ZERO10_A_HC16_1: AND Combination of color and shape using differences of Zernike moment invariants up to tenth order polynomials and differences of color histograms, quantified to 16 classes or bins, with redistribution of boundary values.

All the tools involved in the developed software are free distribution tools, like vs. 2.6.9 Linux operating system, vs. 4.0 of the GCC GNU compiler [18], CVS version of the FFmpeg video stream decoder [19] and vs. 2.6.21 of the LIBXML2 library for processing XML files [20].

5.2 Results analysis

Table 1 shows the recall and precision values as returned by the TRECVID05 team for the cut detection task. It must be noticed that the submitted test for the official runs included a version where very short dissolves were not considered as cuts. Due to the fact that TRECVID05 assumes the opposite, half of the submitted run tests have been removed from this report since it would mean to replicate the results in both cases. We have included in this table our own results for the ZERO10_A_HC16_1 primitive and the results achieved by the color histogram primitive H_C16 using the 2005 test data.

Table 1 shows the influence of the Zernike's polynomials order on the recall and precision measures (labels ZER3, ZER5 and ZER10), obtaining better results when the polynomial degree is higher. Precision value improves with higher order polynomials, since working with a feature vector with more discriminant power means achieving a reduction in the number of false positives detected. On the other hand, this improvement it is not worth when its high computational cost is considered.

About the combination of primitives, it can be observed that the OR technique improves the recall but not the precision. On one hand, the number of true positives is higher, since each primitive contributes with true positives not detected by the other one. On the other hand, each primitive is adding to the combination a number of false positives and false negatives not considered by the other one.

Secondly, introducing a certainty value improves significantly the precision of the techniques, since it is clear a reduction on the number of false positives and negatives. In fact, the best global precision is obtained using this method. By contrast, the recall value lowers down since some of the true positives can have a low certainty value. But it must be noticed how this method doubles the precision value while the reduction of the recall is under a 20%.

Finally, the recall achieved by the AND combination reflects the true positives that both primitives have detected. It is slower since there are some true positives detected by one of the primitives but not by the other one, and also a few of them not detected at all. In this case the precision is greater than the ones achieved by the primitives alone since using this technique the number of false positives and negatives is heavily reduced to only those detected by both primitives.

Decoding and segmentation time has been obtained on a 3GHz Pentium IV processor. Table 2 shows data about the processing complexity of the runs submitted to the TRECVID. The decode time includes a rescaling stage for every frame. It has not been included into the segmentation time because it has been implemented using a general algorithm based on bicubic interpolation without considering any optimization in terms of performance. It can be observed the higher computational load demanded by the primitives involving Zernike invariants, greatly increasing the load demand as the polynomials degree raises.

Figure 4 shows the combination of recall and precision values for the runs shown on Table 1. The figures show the high stability of the methods, since it can be seen that all the values are grouped around the mean values.

6 Conclusions and ongoing work

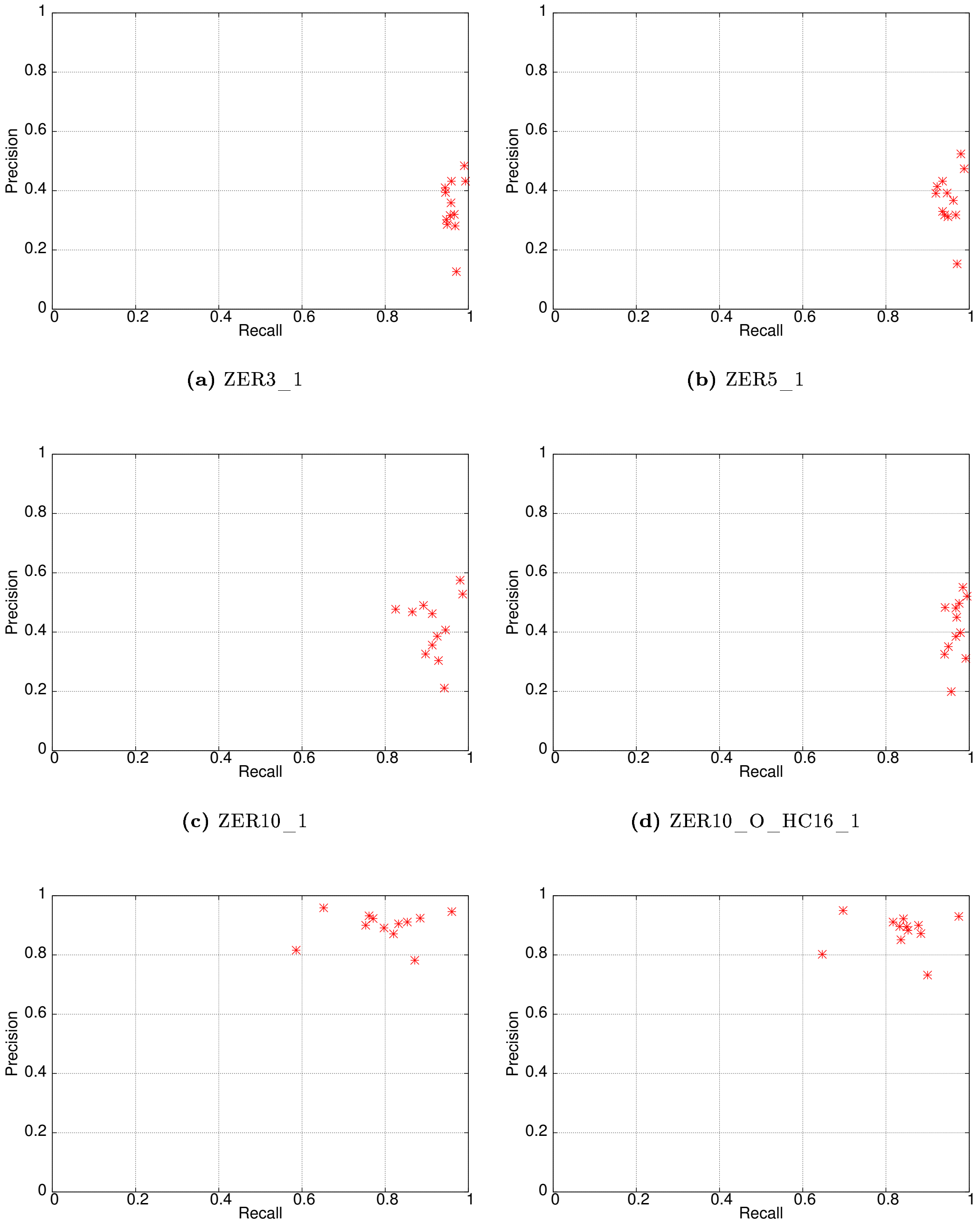
In this paper some automatic shot detection techniques based only on shape and on a combination of shape and color are presented. The shape primitive based on Zernike invariants achieves interesting results using low level order polynomials, diminishing the main drawback of this technique: its high demanding computational load. It can be noticed that all the Zernike primitives achieve recall values over 0.922.

All the experiments combining both primitives, color and shape, show great improvements. The OR technique increased the recall value, achieving the best recall value. When using a certainty value a very significant improvement of the precision was observed, being also the best precision value in all the experiments. Finally, using the AND method allows to obtain the best trade-off between precision and recall values.

Further research on multiresolution histograms, whose initial approach to shot segmentation was presented during TRECVID04, will allow to improve the results. To conclude, the use of other alternatives for combining both primitives will also be studied in the near future.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Education and Science (grant TIC2003-08933-C02) and Government of the Community of Madrid (grant GR/SAL/0940/2004).



(a) ZER3_1

(b) ZER5_1

(c) ZER10_1

(d) ZER10_O_HC16_1

(e) ZER10_O_HC16_UM_1

(f) ZERO10_A_HC16_1

Figure 4: Recall–precision graphs of the implemented primitives.

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, Pablo Toharia, Angel Rodríguez, and Luis Pastor. Automatic video cut detection using adaptive thresholds. In Juan José Villanueva, editor, *Proceedings of the Fourth IASTED International Conference on Visualization, Imaging and Image Processing*, pages 517–522, Marbella, Spain, September 2004. IASTED, ACTA Press. ISBN: 0-88986-454-3, ISSN:1482-7921.

[13] Óscar David Robles Sánchez. *Técnicas de Recuperación por Contenido para Imagen y Vídeo en Arquitectruas Paralelas.* Ph. D. dissertation, Universidad Politécnica de Madrid, Facultad de Informática, 28660 Boadilla del Monte, Madrid, España, December 2004.

[14] Oscar D. Robles, Pablo Toharia, Angel Rodríguez, and Luis Pastor. Towards a content-based video retrieval system using wavelet-based signatures. In M. H. Hamza, editor, *7th IASTED International Conference on Computer Graphics and Imaging - CGIM 2004*, pages 344–349, Kauai, Hawaii, USA, August 2004. IASTED, ACTA Press. ISBN: 0-88986-418-7, ISSN:1482-7905.

[15] A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.

[16] N. K. Kamila, S. Mahapatra, and S. Nanda. Invariance image analysis using modified zernike moments. *Pattern Recognition Letters*, 26(6):747–753, May 2005.

[17] F. Zernike. Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode (Diffraction theory of the cut procedure and its improved form, the phase contrast method). *Physica*, 1:689–704, 1934.

[18] GNU. www.gnu.org

[19] Sourceforge. FFMPEG multimedia system. ffmpeg.sourceforge.net

[20] Gnome Project. Gnome XML C parser and toolkit. www.xmlsoft.org