Video Segmentation by Supervised Learning

G. Cámara Chávez[1,2], M. Cord[1], F. Precioso[1], S. Philipp-Foliguet[1], Arnaldo de A. Araújo[2]
[1]Equipe Traitment des Images et du Signal-ENSEA
6 avenue du Ponceau 95014 Cergy-Pontoise - France

[2]Federal University of Minas Gerais
Computer Science Department
Av. Antônio Carlos 6627 31270-010 - MG - Brazil

Abstract

In most of video shot boundary detection algorithms, proposed in the literature, several parameters and thresholds have to be set in order to achieve good results. In this paper, to get rid of parameters and thresholds, we explore a supervised classification method for video shot segmentation. We transform the temporal segmentation into a class categorization issue. Our approach defines a uniform framework for combining different kinds of features extracted from the video. Our method does not require any pre-processing step to compensate motion or post-processing filtering to eliminate false detected transitions. The experiments, following strictly the TRECVID 2002 competition protocol, provide very good results dealing with a large amount of features thanks to our kernel-based SVM classification method.

1. Introduction

The development of shot boundary detection algorithms was initiated some decades ago with the intention of detecting sharp cuts in video sequences. A vast majority of all works published in the area of content-based video analysis and retrieval are related in one way or another with the problem of shot boundary detection. Indeed, solving the problem of shot boundary detection is one of the principal prerequisites for revealing video content structure in a higher level.

A common approach to detect shot boundaries is computing the difference between two adjacent frames (color, motion, edge and/or texture features) and compare this difference to a preset threshold (threshold-based approach). Del Bimbo [7], Brunelli et al. [4], Lienhart [16] collect extensive reviews of this set of techniques. The main drawback of these approaches lies in detecting different kinds of transitions with a unique threshold. To cope with this problem, video shot segmentation may be seen, from a different perspective, as a categorization task. There have only been a few machine learning approaches proposed to overcome this problem. Boreczky et al. [3] apply HMMs with separate states to model shot cuts, fades, dissolves, pans and zooms. Gunsel et al. [11] consider temporal video segmentation as a 2-class clustering problem ("scene change" and "no scene change") and use K-means to cluster frame differences. Different supervised approaches were proposed by [19], [1] and [2]. Recently Ewerth et al. proposed an unsupervised approach [8].

The work presented in this paper focuses on the exploitation of features based on frame differences (histograms, projection histogram, Fourier-Mellin moments and phase correlation method) for abrupt transition (cut) detection. After the feature extraction step, these features are classified by *Support Vector Machines* (introduced as a machine learning method by Cortes and Vapnik [6]). Furthermore, SVM have been successfully applied in many real world problems and in several areas: text categorization [13], handwritten digit recognition [25] and object recognition [17], etc.

Most of previous works in cut detection consider a low number of features because of computational and classifier limitations. Then to compensate this reduced amount of information, they need pre-processing steps, like motion compensation. Our kernel-based SVM approach can efficiently deal with a large number of features in order to get a robust classification: better handle of illumination changes and fast move problems, without any pre-processing step.

This paper is organized as follows. In section 2, we present the machine learning approach for cut detection used in this work. In section 3, we detail the visual features used for classification and evaluate the similarity measures applied for matching visual information. We present our modified phase correlation feature, in section 4. In section 5, we describe our kernel-based SVM classifier. In section 6, we present the results of the proposed method. In

section 7, we conclude and we present future work.

2. Machine Learning approach

Statistical learning approaches have been recently introduced in multimedia information retrieval context and have been very successful [23]. For instance, discrimination methods (from statistical learning) may significantly improve the effectiveness of visual information retrieval tasks.

The system that we propose in this paper deals with a statistical learning approach for video cut detection. However, our classification framework is specific. Figure 1 shows the steps of the approach. First, the feature extraction process captures different information of each frame. We extract, for every frame in the video stream a feature vector, then a pairwise similarity measure is calculated. We test different distance metrics: $L1$ norm, cosine similarity, histogram intersection and χ^2 distance (see Sec. 4 for more details). Then, each dissimilarity feature vector (distance for each type of feature: color histogram, moments and projection histograms) is used as an input in the classifier. As soon as we use a lot of features, the dimension of the input classification space is high.

With vectors of high dimensionality, artifacts appear, known as the result of the curse of dimensionality [12]. The theory of kernel functions [22], associated with efficient heuristics for classification (as SVM margin maximization) allow to have flexibility and complexity control.

Using a kernel function leads to a set of classification methods. For Pattern Recognition, statistical learning techniques such as nearest neighbors [12], support vector machines, bayes classifiers have been used.

SVM have been successful for many multimedia application problems [13], [17], and we have also previously shown that SVM are highly adapted to the image retrieval context [10]. Thus, we adopted here SVM as classification method. The decision function (previously trained using a data set selected for that purpose) provides as a result the binary labels, i.e., if the frame is detected as a "cut" or "non cut".

The advantage of this approach is that all the thresholds are tuned by the classifier. Thus, the number of features do not represent an issue. Another advantage of the approach is that with many features it is possible to better describe the information content in the frame and avoid the pre-processing step. We denote with subscript $_h$ vectorial/histogram features to discriminate them from scalar features (PC and Var).

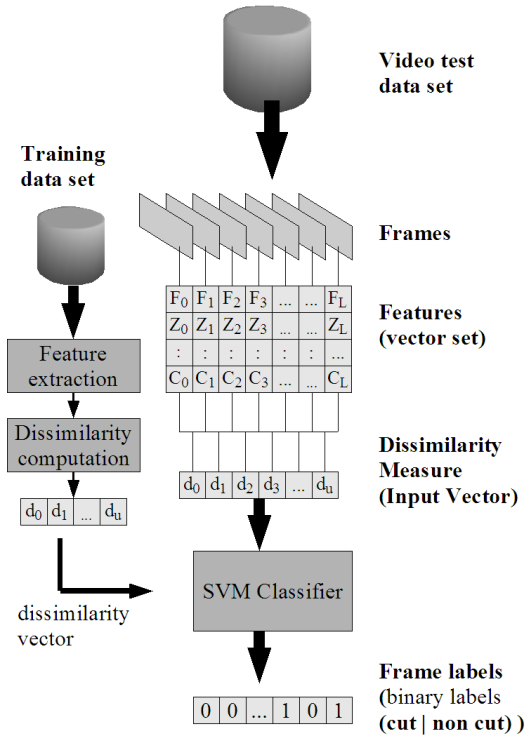


Figure 1. Learning-based Approach for video cut detection. Feature vectors $F_h, Z_h, \ldots C_h$ represent Fourier Mellin moments, Zernike moments, Color histogram (RGB, HSV and R-G), from frame f_t. The other features are detailed in Section 3. $d_t = D(f_t, f_{t+1})$ is the similarity distance for each feature where D is one of the similarity measure detailed in Section 4. The SVM classifier is detailed in Section 5.

3. Visual features

Cuts generally correspond to an abrupt change between two consecutive images in the sequence. Automatic detection is based on the information extracted from the shots (brightness, color distribution, motion, edges, etc.). Cut detection between shots with little motion and constant illumination, is usually done by looking for sharp brightness changes. However, brightness changes cannot be easily related to transition between two shots, in the presence of continuous object motion, or camera movements, or change of illumination. Thus, we need to combine different and more complex visual features to avoid such problems. In the next subsections we will review the main visual features used for shot boundary detection.

3.1. Color Histogram

Let $I(x, y)$ be a color image of size $m \times n$, which consists of three channels $I = (I_1, I_2, I_3)$, the color histogram used here is:

$$h_c(b) = \frac{1}{m \times n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} \begin{cases} 1 & \text{if } I(x,y) \text{ in bin } b \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The color spaces used in this work are the RGB, HSV and opponent color (brightness-independent chromaticities space). Usually the number of bits per channel is set to 2 or 3 in order to reduce sensitivity to noise and slight light, object as well as view changes [16]. In the case of RGB and HSV we consider 2 bits per channel.

The opponent color representation of RGB color space is defined as: $(R+G+B, R-G, B-R-G)$. By choosing this color space, the proposed cut detection algorithm is less sensitive to lighting changes. The advantage of this representation is that the last two chromaticity axes are invariant to changes in illumination intensity and shadows. In our approach we use the second channel of the opponent color space, i.e., $R-G$ and compute a histogram of 64 bins.

These features are stored in vectors denoted RGB_h, HSV_h, $R-G_h, \ldots$

3.2. Shape descriptors

As shape descriptor we use ortogonal moments like Zernike moments [14] and Fourier-Mellin moments [15].

3.2.1. Zernike moments The Zernike moment, of order pq, is defined as :

$$Z_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_0^1 I(\rho,\theta) V_{pq}^*(\rho,\theta) \rho d\rho d\theta \quad (2)$$

where $p = 0, 1, 2, \ldots, \infty$ defines the order, $I(\rho,\theta)$ is the image luminance in polar coordinates (ρ,θ), while q is an integer depicting the angular dependence, or rotation. The Zernike polynomial V_{pq} is a set of complex polynomials which form a complete orthogonal basis set defined on the unit circle and $\{\}^*$ denotes the conjugate in complex domain [14, 27]. Moments of order 5 ($p = 5$, $|q| \leq p$ with $p - |q|$even) are computed for each frame, and arranged in a vector denoted Z_h.

3.2.2. Fourier-Mellin moments U_{pq} is the ortogonal Fourier-Mellin function of order p, q (uniformly distributed over the unit circle) defined as:

$$U_{pq}(\rho,\theta) = Q_p(\rho) e^{-jq\theta}, \quad (3)$$

and the orthogonal Fourier-Mellin moments F_{pq} are defined as:

$$F_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_0^1 I(\rho,\theta) U_{pq}(\rho,\theta) \rho d\rho d\theta \quad (4)$$

where $I(\rho,\theta)$ is the image luminance in polar coordinates (ρ,θ), $q = 0, \pm 1, \pm 2, \ldots$ is the circular harmonic order, the order of the Mellin radial transform is an integer p with $p \geq 0$. For a given degree p and circular harmonic order q, $Q_p(\rho) = 0$ has p zeros. The number of zeros in a radial polynomial corresponds to the capacity of the polynomials to describe high frequency components of the image. Therefore, for representing an image over the same level of quality, the order of p ortogonal Fourier-Mellin is always less than the order of other moments [15]. Moments of order 4 ($p = 4$ and $|q| \leq p$) are computed for each frame, all of them arranged in a vector denoted F_h.

3.3. Projection histograms

Projection is defined as an operation that maps the image luminance into a one-dimensional array called projection histogram [24]. Two types of projection (vertical and horizontal):

$$M_{hor}(y) = \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} I(x,y) dx \quad (5)$$

$$M_{ver}(x) = \frac{1}{y_2 - y_1} \int_{y_1}^{y_2} I(x,y) dy \quad (6)$$

These features are stored in vectors denoted V_h and H_h

3.4. Similarity measures

Here, we describe the similarity measures used for matching visual information. The similarity is determined as a distance between 2 extracted vectors representing one feature (for example Zernike moments: Z_h) or concatenation of several features (for example Zernike moments and color histograms: $\{Z_h, HSV_h\}$). Feature vectors are considered as histograms in terms of similarity measure and thus denoted with the generic name H_t for frame f_t.

The distance usually used is a L_1 norm between feature vectors H_t and H_{t+1}.
The cosine dissimilarity [20] between two vectors is defined as:

$$d_t = D(f_t, f_{t+1}) = \frac{\sum_{j=0}^{u} (H_t(j) \times H_{t+1}(j))}{\sqrt{\sum_{j=0}^{u} H_t(j)} \times \sqrt{\sum_{j=0}^{u} H_{t+1}(j)}} \quad (7)$$

where $H_t(j)$ is $j-$th bin of the vector of the $t-$th frame.
Histogram intersection is defined as:

$$d_t = D(f_t, f_{t+1}) = 1 - \frac{\sum_{j=0}^{u} min(H_t(j), H_{t+1}(j))}{\sum_{j=0}^{u} H_t(j)} \quad (8)$$

Another dissimilarity metric is χ^2:

$$d_t = D(f_t, f_{t+1}) = \sum_{j=0}^{u} \frac{(H_t(j) - H_{t+1}(j))^2}{H_t(j) + H_{t+1}(j)} \quad (9)$$

4. Phase Correlation Method (PCM)

The phase-correlation method [26] measures the motion directly from the phase correlation map (shift in the spatial domain is reflected as a phase change in the spectrum domain). This method is based on block matching: each block r in frame f_t is sought the best match in the neighbourhood around the corresponding block in frame f_{t+1}. In this work a block size of 32×32 was chosen. The PCM for one block is defined as:

$$\rho(r_t) = \frac{FT^{-1}\{\widehat{r_t}(\omega)\widehat{r_{t+1}}^{*}(\omega)\}}{\sqrt{\int |\widehat{r_t}(\omega)|^2 d\omega \int |\widehat{r_{t+1}}(\omega)|^2 d\omega}} \qquad (10)$$

where ρ is the spatial coordinate vector and ω is the spatial frequency coordinate vector, $\widehat{r_t}(\omega)$ denotes the Fourier transform of block r_t, FT^{-1} denotes the inverse Fourier transform and $\{\}^*$ is the complex conjugate.

By applying a high-pass filter and performing normalised correlation this method is robust to global illumination changes [18]. Porter [18] suggest the use of the maximum correlation value as a measure for each block, but one problem with this measure is that we do not have information of the neighbors of the maximum correlation value. Instead of using that measure, we propose the use of the entropy E_r of the block r as the *goodness-of-fit* measure for each block. The entropy give us global information of the block, not only information for a single element of the block.

The similarity metric PC is defined by the median of all block entropies instead of the mean to prevent outliers [18].

$$PC = \text{median}(E_r) \qquad (11)$$

Although, the PC feature is particularly relevant in presence of illumination changes, it provides false positive cuts for "black" frames due to Mpeg-1 artifacts. In order to overcome this limitation, we add the illumination variance (Var). Indeed, two "black" frames PC will be high like for non-similar images while variance will be little in the first case and high in the second. Indeed, the PC feature of two successive "black" frames will be high like in case of two non-similar frames while variance will allow us to discriminate these configurations.

5. Cut / non cut classification

There are some learning approaches that use SVM as classifier. More recently, Qi et al. [19] transform the temporal segmentation into a multi-class categorization. For the classification task they compare different binary classifiers: $k-$nearest-neighbor classifier (KNN), the Naïve Bayes probabilistic classification, and SVM. Since its creation in 2001 TRECVID [1] has become the reference framework to propose and compare new approaches. IBM system [2] consists of extraction modules for local and global visual features. The algorithm is based on a finite state machine and the features are classified by a SVM. R. Ewerth and B. Freisleben [8] propose an unsupervised learning approach based on a sliding estimation window and an adequate measure of clustering quality. Adcock et al. [1] present an approach combining pairwise similarity and supervised classification, they used a KNN. Regarding the increase of classification methods proposed for TRECVID and the quality of their results, these approaches appear promising for the task of shot boundary detection. Based on these successful experiences we adopted a machine learning approach.

The classification problem can be restricted to a two-class problem. The goal is, then, to separate the two classes with a function induced from available examples. We hope to produce, hence, a classifier that will properly work on unknown examples, i.e. which generalises efficiently the classes defined from the examples.

The SVM have been developed as a robust tool for classification and regression in noisy and complex domains as multimedia retrieval [17, 23]. SVM can be used to extract valuable information from data sets and construct fast classification algorithms for massive data. Another important characteristic of the SVM classifier is to allow a non-linear classification without requiring explicitly a non-linear algorithm thanks to kernel theory. We adopted SVM with kernel framework that perfectly matches with our binary classification problem with non linear high dimensional data.

In kernel framework data points may be mapped into a higher dimensional feature space, where a separating hyperplane can be found. We can avoid to explicitly compute the mapping using the kernel trick which evaluates similarities between data $K(d_t, d_s)$ in the input space.

Common kernel functions are: linear, polynomial, gaussian radial basis, gaussian with χ^2 distance (Gauss-χ^2) $K(d_t, d_s) = e^{-\chi^2(d_t,d_s)/2\sigma^2}$ and triangular kernel [9] $K(d_t, d_s) = -||d_t - d_s||$. Each kernel function results in a different type of decision boundary.

Our kernel-based SVM approach can thus efficiently deal with a large number of features in order to get a robust classification.

6. Experimentation

6.1. Data set

The training set consists of a single video of 9078 frames (5mins. 2 secs.) with 128 cuts and 8950 non cuts. This video

1 A video retrieval algorithm competition

Run	Features
1	HSV_h, F_h, Z_h, H_h, PC, Var
2	$R-G_h$, HSV_h, RGB_h, F_h, Z_h, PC, Var
3	$R-G_h$, HSV_h, RGB_h, F_h, H_h, PC, Var
4	HSV_h, RGB_h, F_h, Z_h, PC, Var
5	HSV_h, RGB_h, F_h, Z_h, H_h, PC, Var
6	RGB_h, F_h, Z_h, V_h, PC, Var
7	RGB_h, F_h, Z_h, V_h, H_h, PC, Var
8	HSV_h, RGB_h, F_h, Z_h, V_h, H_h, PC, Var
9	$R-G_h$, HSV_h, RGB_h, F_h, Z_h, H_h, PC, Var
10	$R-G_h$, HSV_h, RGB_h, F_h, Z_h, H_h, V_h, PC, Var

Table 1. Combination set of visual features used in our tests.

is captured from a brazilian TV-station and is composed by a segment of commercials. The training video was labeled manually by ourselves.

The test set used in our experiments is TRECVID-2002 Video Data Set (the only set that the ground truth is publicly available). The shot boundary test collection contains 4 hours and 51 minutes of video. The video are mostly of a documentary/educational nature, but very varied in age, production style, and quality. At a total, there were 18 videos in MPEG-1 with a total size of 2.88 gigabytes. For all videos, shot segmentation reference data had been manually constructed by NIST.

We strictly follow the TRECVID-2002 protocol in our tests. We run our algorithm on all the TRECVID test set and provide the mean precision and the mean recall obtained. We can provide up to 10 different runs (10 different choices of parameters, features or kernels). We use the precision, recall and $F1$ statistics defined in TRECVID protocol:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (12)$$

6.2. Features

The nomenclature used for the features is as follows: RGB_h, HSV_h and $R-G_h$ color histograms, Zernike (Z_h) and Fourier-Mellin (F_h) moments , Horizontal (H_h) and Vertical (V_h) projection histograms, Phase correlation (PC) and Variance (Var). In Table 1, we present the visual feature vectors used in our tests.

Figure 2 shows the precision/recall measure for all the sets presented in Table 1. Each set of features was tested with each kernel function (linear, polynomial, Gauss-$L2$, Gauss-χ^2 and triangle), thus we have 5 values for each set in the figure. We can see in the figure that the best performance are executed using (b) cosine dissimilarity and (c) histogram intersection, where we have precision and recall values close to one. We made many experiments, and we always found the best results using these two dissimilarities. In the case of (a) $L1$ norm and (d) χ^2 distance we got a better precision but the recall is not as good as in (b) and (c).

Learning support is robust since with training sets from different camera, from different compress format, coding, from different country, situation, the features keep being relevant and stable to detect cuts in different context and environment.

6.3. Optimization of kernel functions

We use a SVM classifier and train it with different kernels: linear, polynomial, gaussian with L2 and χ^2 distance, and triangular.

We conducted numerous experiments that provide interesting and meaningful contrast. Table. 2 shows the recall, precision and $F1$ measures for the three best similarity measures for each kernel function, also we present the dissimilarity distance used for comparing the feature vectors and the features that were used in each run. The Gaussian-χ^2 kernel provides the best results over all the other kernel functions. Thus, our evaluation of kernels functions confirms that when distributions are used as feature vectors, a Gaussian kernel gives excellent results in comparison to distance-based techniques [10].

6.4. Optimization of training set

In order to reduce the number of support vectors and decrease the time consumed for training and testing we reduce our training set. Instead of using the 5 min. video we segment it and train our classifier with a 2 min. video that contains 50 cuts. The performance of our system maintains its accuracy with the advantage that the steps of training and testing are very fast. In Table 3 we show the recall, precision and $F1$ statistics using seven different feature sets. The choice for kernel is the Gaussian-χ^2 (as it is shown in our experiments it execute the best perform), thus all runs were executed using this kernel and the dissimilarity measure used was the cosine metric. We can see that the performance is still the same for all the runs.

6.5. TRECVID competition

In Table 4 we show the performance of our system. All this results, the best ones, are obtained using the χ^2 kernel. We present the recall and precision and its respective variance. The small values of variance shows the stability of our system. In Figure 3(a) we show the results that were obtained in the official contest of TRECVID-2002 and compare it with the results of our ten runs Figure 3(b). As shown

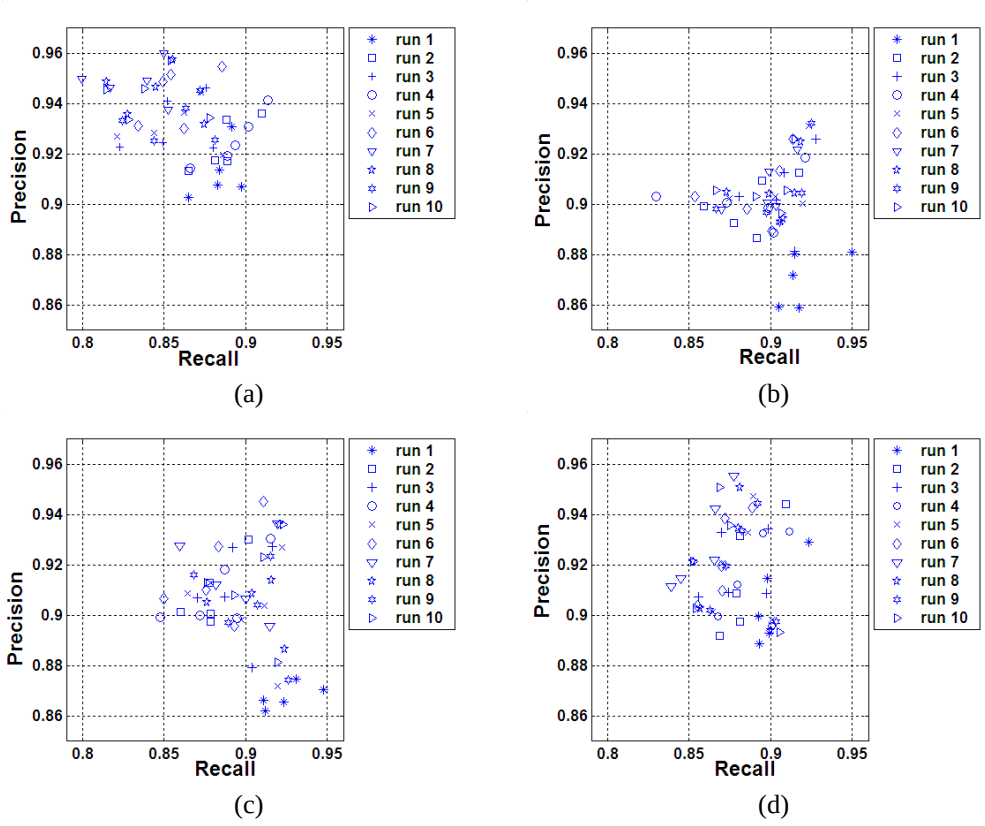


Figure 2. Precion/Recall measure for the ten feature sets using different dissimilarity measures. In each figure we present 5 values for each set, which represent the response values for each kernel. (a) $L1$ norm (b) cosine dissimilarity, (c) histogram intersection and (d) χ^2 distance. ten runs results

in the figure the accuracy and robustness of our approach is very efficient. Hence, the capacity of generalisation of our classifier is proven and the combination of the selected features performs good results without any pre-processing or post-processing.

7. Conclusion and future works

This paper considers cut detection from a supervised classification perspective. Previous detecting cut classification approaches consider few visual features because of computational limitations. As a consequence of this lack of visual information, these methods need pre-processing and post-processing steps, in order to simplify the detection in case of illumination changes, fast moving objects or camera motion. We are actually combining the cut detection method with our content- based search engine previously developed for image retrieval [5] in order to carry out an interactive content-based video analysis system. The kernel-based SVM classifier can deal with large feature vectors. Hence, we combine a large number of visual features and avoid any pre-processing or post-precessing step. We present a supervised statistical learning approach, requiring a small training set. Thus, we do not have to set any threshold as many methods proposed in the framework

Run	Recall	σ_{recall}	Prec.	$\sigma_{\text{prec.}}$	Diss. meas
1	0.929	0.004	0.923	0.010	χ^2 dist.
2	0.944	0.003	0.909	0.014	χ^2 dist.
3	0.926	0.003	0.928	0.007	cos
4	0.941	0.003	0.914	0.009	L1
5	0.931	0.003	0.924	0.007	cos
6	0.945	0.003	0.911	0.007	Hist.Int.
7	0.936	0.004	0.919	0.008	Hist.Int.
8	0.936	0.004	0.921	0.009	Hist.Int.
9	0.932	0.003	0.925	0.007	cos
10	0.936	0.005	0.923	0.007	Hist.Int.

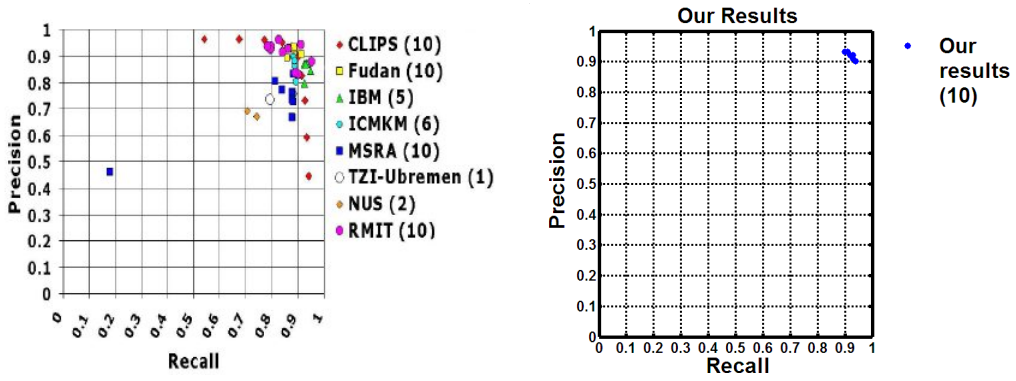
Table 4. Performance of our system with $\chi 2$ kernel function

Kernel	Recall	Precision	F1	Diss. Measure	Features
Linear	0.92	0.90	0.91	Cos.	$R-G_h, HSV_h, RGB_h, F_h, V_h$, Var, PC
	0.92	0.91	0.91	Cos.	$HSV_h, RGB_h, F_h, Z_h, V_h$, Var, PC
	0.92	0.90	0.91	Cos.	$R-G_h, HSV_h, RGB_h, F_h, Z_h, V_h$, Var, PC
Poly	0.92	0.90	0.91	$L1$	$R-G_h, HSV_h, RGB_h, F_h, Z_h$, Var, PC
	0.91	0.92	0.91	$L1$	$R-G_h, HSV_h, F_h, Z_h, V_h$, Var, PC
	0.93	0.90	0.92	$L1$	HSV_h, RGB_h, F_h, Z_h, Var, PC
Gauss-$L2$	0.91	0.90	0.91	Hist.Int.	$HSV_h, RGB_h, F_h, Z_h, V_h, H_h$, Var, PC
	0.92	0.90	0.91	Hist.Int.	$R-G_h, RGB_h, F_h, Z_h, V_h, H_h$, Var, PC
	0.92	0.90	0.91	Cos.	$HSV_h, RGB_h, F_h, V_h, H_h$, Var, PC
Gauss-$\$chi^2$	0.93	0.92	0.93	Cos.	$R-G_h, HSV_h, RGB_h, F_h, Z_h, H_h$, Var, PC
	0.94	0.92	0.93	Cos.	$HSV_h, RGB_h, F_h, Z_h, V_h, H_h$, Var, PC
	0.94	0.92	0.93	Cos.	$R-G_h, HSV_h, RGB_h, F_h, Z_h, V_h, H_h$, Var, PC
Triangle	0.91	0.92	0.92	Cos.	$HSV_h, RGB_h, F_h, V_h, H_h$, Var, PC
	0.92	0.91	0.92	Hist.Int.	$R-G_h, HSV_h, RGB_h, F_h, Z_h, V_h, H_h$, Var, PC
	0.92	0.92	0.92	Cos.	$R-G_h, HSV_h, RGB_h, F_h, V_h, H_h$, Var, PC

Table 2. Measure performance for each kernel function.

Complete Train Set 128			Reduced Train Set 50			
Recall	Prec.	F1	Recall	Prec.	F1	Features
0.92	0.92	0.92	0.90	0.93	0.92	HSV_h, Z_h, H_h, Var, PC
0.92	0.92	0.92	0.91	0.93	0.92	HSV_h, V_h, H_h, Var, PC
0.93	0.90	0.92	0.93	0.91	0.92	HSV_h, RGB_h, F_h, H_h, Var, PC
0.93	0.91	0.92	0.92	0.92	0.92	HSV_h, Z_h, V_h, H_h, Var, PC
0.94	0.90	0.92	0.93	0.91	0.92	$R-G_h, HSV_h, F_h, H_h$, Var, PC
0.95	0.90	0.93	0.93	0.91	0.92	$HSV_h, RGB_h, F_h, Z_h, H_h$, Var, PC
0.94	0.90	0.92	0.93	0.91	0.92	$R-G_h, HSV_h, RGB_h, F_h, Z_h, H_h$, Var, PC

Table 3. Comparison of performance for 7 feature sets using all training set videos and the reduced training set videos.



(a) (b)

Figure 3. Precion/Recall measure of performance. (a) show the official results for TRECVID 2002 [21], (b) show our ten runs results

of TRECVID. We compare our algorithm to the latest results publicly available. Our method shows excellent performance on the 2002 TREC Video Track Shot Classification Task in terms of precision and recall. To confirm the efficiency of our approach, we are going to participate to the TRECVID-2006 competition. The next step is to extend our algorithm for gradual transition detection. For that purpose new features will be necessary. This will not be an issue for our kernel-based algorithm which can deal with high order features. We expect our learning-based approach be able to detect cuts and gradual transitions.

8. Acknowledgments

The authors are grateful to MUSCLE Network of Excellence, CNPq and CAPES for the financial support of this work.

References

[1] J. Adcock, A. Gingensohn, M. Cooper, T. Liu, L. Wilcox, and E. Rieffel. Fxpal experiments for trecvid 2004. In *TREC Video Retrieval Evaluation Online Proceedings: TRECVID 2004*, 2004.

[2] A. Amir, J. Argillander, M. Berg, S.-F. Chang, M. Franz, W. Hsu, G. Uyengar, J. Kender, L. Kennedy, C. Lin, M. Naphade, A. Natsev, J. Smith, J. Tesic, G. Wu, R. Yan, and D. Zhang. Ibm research trecvid-2004 video retrieval system. In *TREC Video Retrieval Evaluation Online Proceedings: TRECVID 2004*, 2004.

[3] J. Boreczky and L. Wilcox. A hidden markov model framework for video segmentation using audio and image features. In *ICASSP'98*, volume 6, pages 3741–3744, 1998.

[4] R. Brunelli, O. Mich, and C. Modena. A survey on the automatic indexing of video data. *Journal of Visual Communication and Image Representation*, 10:78–112, 1999.

[5] M. Cord, P. Gosselin, and S. Philipp-Foliguet. Stochastic exploration and active learning for image retrieval. *Image and Vision Computing journal*, accepted for publication, Jan 2006.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] A. del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, San Francisco, California, 1999.

[8] R. Ewerth and B. Freisleben. Video cut detection without thresholds. In *Proc. of 11th Workshop on Signals, Systems and Image Processing*, pages 227–230, Poznan, Poland, 2004. PTETiS.

[9] F. Fleuret and H. Sahbi. Scale-invariance of support vector machines based on the triangular kernel. In *3th International Workshop on Statistical and Computational Theories of Vision*, 2003.

[10] P. Gosselin and M. Cord. A comparison of active classification methods for content- based image retrieval. In *International Workshop on Computer Vision meets Databases (CVDB), ACM Sigmod*, pages 51–58, Paris, France, June 2004.

[11] B. Gunsel, A. Fernan, and A. Tekalp. Temporal video segmentation using unsupervised clustering and semantic object tracking. *Journal of Electronic Imaging*, pages 592–604, 1998.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The Element of Statistical Learning*. Springer, 2001.

[13] T. Joachims. Text categorization witt support vector machines. In *Proceedings of the European Conference on Machine Learning*, 1998.

[14] C. Kan and M. Srinath. Combined features of cubic b-spline wavelet moments and zernike moments for invariant pattern recognition. In *International Conference on Information Technology: Coding and Computing.*, pages 511–515, 2001.

[15] C. Kan and M. Srinath. Invariant character recognition with zernike and orthogonal fourier-mellin moments. *Pattern Recogntion*, 35:143–154, 2002.

[16] R. Lienhart, C. Kuhmunch, and W. Effelsberg. On the detection and recognition of television commercials. *IEEE International Conference on Multimedia Computing and Systems '97. Proceedings*, pages 509 – 516, 1997.

[17] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the International Conference on Computer Vision*, 1998.

[18] S. V. Porter, M. Mirmehdi, and B. T. Thomas. Temporal video segmentation and classification of edit effects. *Image and Vision Computing*, 21(13-14):1097–1106, December 2003.

[19] Y. Qi, T. Liu, and A. Hauptmann. Supervised classification of video shot segmentation. In *IEEE Conference on Multimedia & Expo (ICME'03)*, Baltimore, MD, July 6-9 2003.

[20] G. Salton. *Automatic Text Processing*. Addison-Wesley Longman Publishing, 1989. Chapter 9.

[21] A. Smeaton and P. Over. The trec-2002 video track report. In *The Eleventh Text REtrieval Conference (TREC 2002)*, 2002. `http://trec.nist.gov//pubs/trec11/papers/VIDEO.OVER.pdf`.

[22] A. Smola and B. Scholkopf. *Learning with kernels*. MIT Press, Cambridge, MA., 2002.

[23] S. Tong. *Active Learning: Theory and Applications*. PhD thesis, Stanford University, 2001.

[24] O. Trier, A. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29:641–662, 1996.

[25] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[26] J. Z. Wang. Methodological review - wavelets and imaging informatics : A review of the literature. *Journal of Biomedical Informatics*, pages 129–141, July 2001. Avaliable on `http://www.idealibrary.com`.

[27] K. Whoi-Yul and K. Yong-Sung. A region-based shape descriptor using zernike moments. *Image Communication*, 16(95-102), 2000.