SHOT BOUNDARY DETECTION AT TRECVID 2006

G. Cámara Chávez[1,2]*, F. Precioso*[1]*, M. Cord*[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

The first step for video-content analysis, content-based video browsing and retrieval is the partitioning of a video sequence into shots. A shot is the fundamental unit of a video, it captures a continuous action from a single camera and represents a spatio-temporally coherent sequence of frames. Thus, shots are considered as the primitives for higher level content analysis, indexing and classification. Although many video shot boundary detection algorithms have been proposed in the literature, in most approaches, several parameters and thresholds have to be set in order to achieve good results. In this paper, we present a robust learning detector of shot boundaries without any threshold to set nor any pre-processing step to compensate motion or post-processing filtering to eliminate false detected transitions. Our experiments provide very good results dealing with a large amount of features thanks to our kernel-based SVM classifier method.

1. INTRODUCTION

The development of shot boundary detection algorithms was initiated some decades ago with the intention of detecting abrupt transitions (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.

Shots can be divided in two groups: abrupt transitions and gradual transitions. Gradual transitions (GT) include camera movements: panning, zooming, tilting and video editing effects: fade-in, fade-out, dissolve and wipe.

A common approach to detect abrupt transitions 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 [1], Brunelli et al. [2], Lienhart [3] collect extensive reviews of this set of techniques. The main drawback of these approaches lies in detecting different kind of transitions with a unique threshold. To cope with this problem, video shot segmentation can be seen, from a different perspective, as a categorization task. The detection of abrupt transitions between two shots was the most extensively studied area in shot boundary detection, nowadays gradual transition detection becomes the new challenge.

GT detection could not be based on the same assumption of cut detection, i.e., high similarity between frames corresponding to the same shot and low similarity between frames corresponding to two successive shots, since this similarity is also high in gradual transitions. The visual patterns of many GT are not as clearly or uniquely defined as that of abrupt transitions.

Among the most commonly used features for gradual transitions, we must mention intensity variance statistics [4, 5, 6]. In [7], Zhang *et al.* used a twin threshold mechanism based on histogram difference metric. Zabih *et al.* [8] have used a measure based on the number of edges changes for detecting editing effects, also for cut detection. Other methods are based on correlation in frame differences[9, 10].

Recently machine learning approaches were proposed to overcome the problem of shot boundary detection. [11] apply HMMs with separate states to model shot cuts, fades, dissolves, pans and zooms. Adcock et al. [12] combines pairwise dissimilarity analysis, kernel correlation and unsupervised multi-class clustering. Gunsel et al. [13] consider temporal video segmentation as a 2-class clustering problem ("scene change" and "no scene change") and use K-means to cluster frame differences. Qi et al. [14] transform the temporal segmentation into a multi-class categorization.

We propose to use a supervised classification technique to determine both kinds of transitions, in a hierarchical scheme composed of two stages. We first extract features computed on frame differences in order to detect abrupt transitions. The features are classified by a kernel-based SVM (Support Vector Machines) classifier, because of its well-known performances in statistical learning information retrieval [15]. Indeed, the SVM is a two-class classifier able to separate between cuts and non-cuts, after training with a selected data-set. Moreover with the use of kernel functions, it can efficiently deal with a large number of features. With many features it is possible to better describe the information included in the shot and to better handle illumination changes and fast movement problems. Thus the pre-processing steps are not necessary anymore. Once the video sequence is segmented into sharp-cut-free segments, we extract new features in each segment in order to detect possible gradual transitions without using any sliding window like most of the authors do [14].

This paper is organized as follows. In section 2, we present our machine learning approach for cut detection. In section 3, we detail our gradual transitions detection algorithm. In section 4, we describe our kernel-based SVM classifier. In section 5, we present the results of the proposed method. In section 6, we conclude and we present future work.

2. CUT DETECTION

Statistical learning approaches have been recently introduced in multimedia information retrieval context and have been very successful [16]. 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 use χ^2 distance as a dissimilarity metric. 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.

Using a kernel function leads to a set of classification methods. For Pattern Recognition, statistical learning techniques such as nearest neighbors [17], support vector machines, bayes classifiers have been used. We have previously shown that the SVM classification method is highly adapted to the multimedia retrieval context [18]. Thus, we use 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. The choice of SVM as a classifier is due to the well known performance in statistical learning information retrieval.

2.1 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. The features used in this work are:

1. *Color Histograms*: The color spaces used in this work are the RGB, HSV and opponent color (brightness-independent chromaticities space). In the case of RGB and HSV we consider 2 bins 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.
   These features are stored in vectors denoted RGBh, HSVh, R-Gh,...
2. *Shape descriptors*
   - *Zernike moments*: The Zernike moment, of order pq, is defined as :

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

     where $p = 0, 1, 2, \ldots, \infty$ defines de order, $I(\rho,\theta)$ is the image 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 [19, 20]. Moments of order 5 ($p = 5$, $p - |q| =$ even and $|q| \leq p$) are computed for each frame, and arranged in a vector denoted Z_t.
   - *Fourier-Mellin moments*: U_{pq} is the ortogonal Fourier-Mellin function of order p, q (uniformly distribute over the unit circle) defined as:

     $$U_{pq}(\rho,\theta) = Q_p(\rho) e^{-jq\theta}, \quad (2)$$

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

     $$F_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_0^1 I(\rho,\theta) U_{pq}(\rho,\theta) \rho d\rho d\theta \quad (3)$$

     where $I(\rho,\theta)$ is the image 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

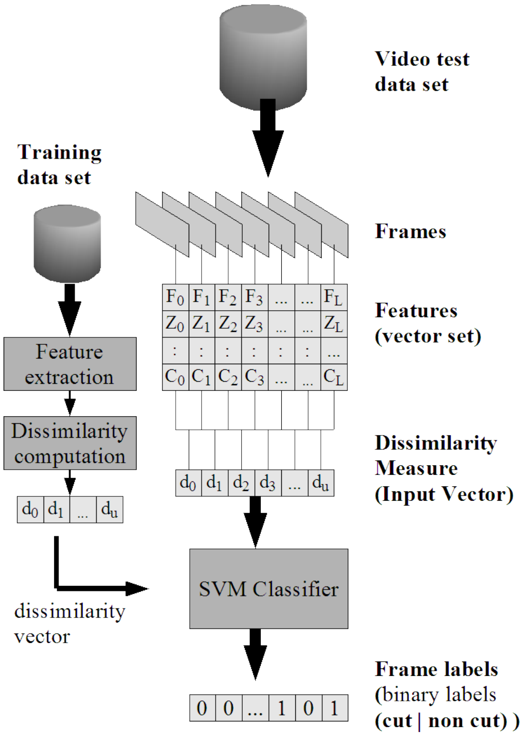
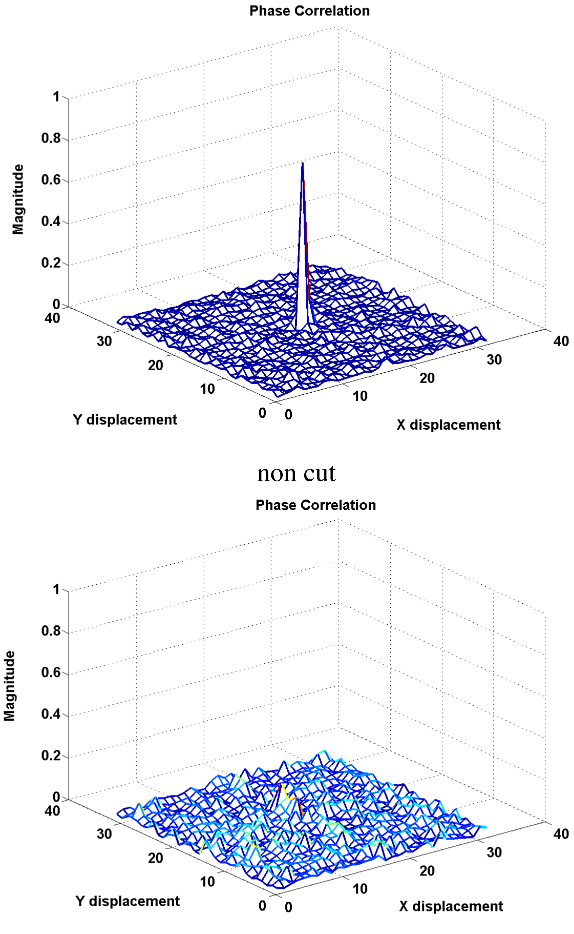


Figure 1: **Learning-based Approach for video cut detection**. Feature vectors $F_t, Z_t, \ldots C_t$ represent Fourier Mellin moments, Zernike moments, Color histogram, from frame f_t. The other features are detailed in Section 3. $d_t = D(f_t, f_{t+1})$ is the similarity distance for each feature where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

with $p \geq 0$. For a given degree p and circular harmonic order q, $Q_p(\rho) = 0$ has p zeros.

3. *Projection histograms*: Projection is defined as an operation that maps a image into a one-dimensional array called projection histogram [21]. Two types of projection (vertical and horizontal). These features are stored in vectors denoted V_h and H_h.
4. *Phase Correlation Method (PCM)*: The phase-correlation method [22] 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}. When one frame is the translation of the other, the PCM has a single peak at the location corresponding to the translation vector. When there are multiple objects moving, the PCM tends to have many peaks, see Figure 2.



non cut

cut cut

Figure 2: Phase correlation.

In this work a block size of 32 × 32 was chosen. The PCM for one block is defined as:

$$\rho(r_t) = \frac{FT^{-1}\{\widehat{r_t}(\omega)\widehat{r_{t+1}}^{*}(\omega)\}}{\sqrt{\int |\widehat{r_t}(\omega)|^2 d\omega \int |\widehat{r_{t+1}}(\omega)|^2 d\omega}} \qquad (4)$$

where ρ is the spatial coordinate vector and ω is the spatial frequency coordinate vector, $\widehat{r_t}(\omega)$ denote 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 [23]. Porter [23] 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 M_t is defined by the median of all block entropies instead of the mean to prevent outliers [23].

$$M_t = \text{median}(E_r) \qquad (5)$$

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.

3. DISSOLVE DETECTION

In [7], Zhang *et al.* used a twin threshold mechanism based on histogram difference metric. Frame differences are accumulated as long as inter-frame difference is above a lower threshold but smaller than a higher threshold. When the accumulated differences exceed the higher threshold, a gradual transition is defined. Camera or object motions may result in a sustained increase in the inter-frame difference same as gradual transitions and cause false detection. Zabih *et al.* [8] have used a measure based on the number of edges changes for detecting editing effects, also for cut detection. This method requires global motion compensation before computing dissimilarity. Low precision rate and time-consuming are the drawbacks of this technique. Another feature that is commonly used for dissolve detection is intensity variance. During a dissolve transition, the intensity curve forms a downwards-parabolic shape. In [4], Alattar proposed a variance-based approach, many other researchers have used this feature to build their dissolve detectors [5, 6]. Alattar [4] proposed to take the second derivative of intensity variance, and then check for two large negative spikes. Again object/camera motion and noise make difficult the dissolve detection (spikes were not to pronounced due to motion and noise). Troung *et al.* [6] proposed an improved version with more constraints. Won *et al.* [24] proposed a method based on the analysis of a dissolve modeling error that is the difference between an ideally modeled dissolve curve without any correlation and an actual variance curve with a correlation. Other researches based on correlation are [9, 10]. Nam and Tewfik [25] use B-spline polynomial curve fitting technique to detect dissolves. The main drawback of these approaches lies in detecting different kind of transitions with a unique threshold. We want to be rid of the threshold setting as much as possible.

In this second step (gradual transition detection), we compute illumination variance and global edge information, with the Effective Average Gradient (EAG), looking for specific shapes of these curves that characterize dissolve. We detect the candidate regions, then process a verification based on our Double Chromatic Difference (DCD).

3.1 Dissolve modeling error

Won *et al.* [24] demonstrated the effect of correlation between neightbor scenes. Early researches in dissolve detection based their methods in the characteristics of an ideal model without any correlation between neighbor scenes. However, in real sequences, there is often a correlation between neighbor scenes, which affects the dissolve detection. This correlation must be taken into account for the precise detection of a dissolve.

The dissolve modeling error [24] is the difference between an ideal dissolve for region $[p,q]$ and the actual variance curve. At the center of a dissolve, the dissolve modeling error is proportional to the correlation [24].

If a correlation c is defined in the region $[p,q]$, the maximum dissolve modeling error M_{max_c}, becomes $\frac{\sigma_p \sigma_q c}{2}$. A dissolve is detected if the maximum dissolve modeling error M_{max} is less than M_{max_c}, this region can be identify as a dissolve with a correlation of less than c. Hence, the maximum dissolve error M_{max_c} with correlation c becomes an adaptative threshold determinated by the characteristics of each region, where c is the target correlation.

3.2 Variance sequence

In this subsection, we describe the first feature used for finding candidate regions. The candidate regions are extracted using the first and second derivatives of the variance curve.

3.3 Effective average gradient (EAG)

In this subsection, we describe the second feature used for finding candidate regions. The candidate regions are extracted using the first and second derivatives of the effective average gradient curve (EAG).

The local edge magnitude can be computed with

$$G^2 = (G_x^2 + G_y^2) \tag{6}$$

where G_x is the gradient on horizontal direction and G_y is the gradient on vertical direction. The gradient magnitude of image sequence during dissolve also show parabolic shape.

The edge count (EC) of an image is the total number of pixels with non-zero gradient values:

$$EC(t) = |F(G) > 0|_{(x,y)} \tag{7}$$

where $|.|$ denotes de cardinality and $F(.)$ is a thresholding function. EC results more accurately for predicting fades and dissolves, as compared to color histogram, frame difference, and motion vector.

The EAG is simply defined as the average edge intensity of a given image:

$$EAG(t) = \frac{\sum_{(x,y)} F(G)}{EC(t)} \tag{8}$$

where G is the local edge magnitude, [26].

3.4 Dissolve filtering

Candidate regions are only identified using on analysis of characteristics of first and second derivative of temporal evolution curves of both variance and EAG, i.e., searching a downward parabola.

3.4.1 Verification of candidate region

Some of the candidate regions may include parabolas corresponding to false dissolve caused by object and camera motion. Therefore, a parabola corresponding to a true dissolve must be distinguished using other dissolve characteristics. Candidate regions are verified with on the modeling error [24].

Figure 3 shows a flow chart for verifying the dissolve region. For each candidate region, the maximum dissolve modeling error D_{max_c} between a dissolve model with a given target correlation c and an ideal dissolve model with no correlation is estimated with variances at the start and end points of each candidate region and the given target correlation c. Then D_{max} becomes the adaptive threshold to verify each candidate region as a dissolve.

The maximum dissolve modeling error D_{max} in each candidate is defined by the difference between the variance σ^2_{center} at the center of each candidate region and the variance $\tilde{\sigma}^2_{center}$ at the center of an ideal dissolve model. Then, the double chromatic difference (DCD) is computed for each regions accepted as possible dissolves

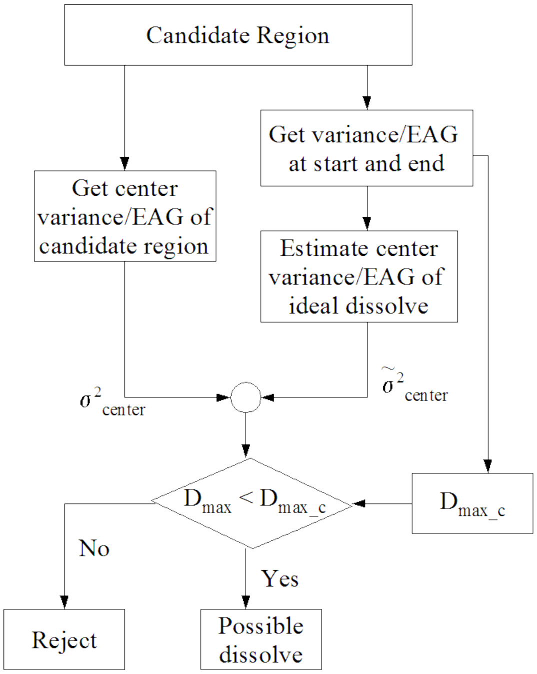

Candidate Region
Get variance/EAG at start and end
Get center variance/EAG of candidate region
Estimate center variance/EAG of ideal dissolve
σ^2_{center}
$\tilde{\sigma}^2_{center}$
$D_{max} < D_{max\_c}$
D_{max_c}
No
Yes
Reject
Possible dissolve

Figure 3: Flow chart for verifying dissolve region

3.5 Modified Double Chromatic Difference

We refine the dissolve detection obtained with *dissolve modeling error* using a modification of the DCD test proposed by Yu *et al.* [27]. The feature can identify dissolve from zoom, pan and wipe. The DCD of frame f_t of a moving image sequence is thus defined as the accumulation of pixel-wised comparison between this average and the intensity of frame $f(x,y,t)$, where $f(x,y,t)$ is a frame in the possible segment of dissolve.

$$DCD(t) = \sum_{x,y} F(|\frac{f(x,y,t_0)+f(x,y,t_N)}{2} - f(x,y,t)|) \quad (9)$$

where $t_0 \leq t \leq t_N$, t_0 and t_N define the starting point and ending frames of a dissolve period. $F(.)$ is a thresholding function.

Ideally, there exists a frame $f(x,y,t)$, where

$$f(x,y,t) = \frac{f(x,y,t_0)+f(x,y,t_N)}{2} \quad (10)$$

We propose a modification of this well known descriptor reducing highly the complexity of its computation. Indeed, we use projection histograms [21] (1D) instead of the frame (2D). Projection histograms allow us not only to reduce the size of data concerned with DCD test but also to preserve color and spatial information. For out modified DCD, the formulation Eq. (9) remains the same if $f(x,y,t)$ represents projection histogram.

3.6 Visual features

Before we describe the features we use in our approach it is important to remember that when we said center of region, we are talking about the position, along the curves of variance and EAG, with the lowest value in the interval (candidate region). The features used are:

1. *Ddata*: different information extracted from the dissolve region, the features used are:
   (a) 2 correlation values : one between frames at the beginning and the ”center”, the other between the ”center” and the end of the dissolve segment,
   (b) 2 color histogram differences : one between frames at the beginning and the ”center”, the other between the ”center” and the end of the dissolve segment,
   (c) correlation by blocks of interest in the sequence: this feature is computed only on the target intervals and use the dissolve descriptor [28].
2. DCD features: the DCD curve (2D) is approximated by a parabola. The first DCD feature is the quadratic coefficient of this parabola [29]. The second is the “depth” of the parabola, defined as the height difference taken at dissolve segment boundaries and at the “center” [5]. The “center” position is defined by the maximal value of the DCD curve inside the dissolve segment.
   Some characteristics are computed from the DCD: curve fitting with degree of two is used to match with the parabola generated by DCD in the case of a true dissolve and the “deep” of the parabola based in the ratio of the boundaries and the “center” of the parabola. The performance is evaluated by the estimated quadratic coefficient $\hat{B_2} > 0$, see [29] and the ratio ψ [5]

$$\psi(i) = \left\{ \begin{array}{l} 1 - \frac{min(DCD_i(m),DCD_i(N))}{max(DCD_i(m),DCD_i(N))}, \text{ if } R \leq 0 \\ 1 - \frac{min(DCD_i(m),DCD_i(0))}{max(DCD_i(m),DCD_i(0))}, \text{ if } R > 0 \end{array} \right. \quad (11)$$

   where $R = |DCD_i(m) - DCD_i(N)| - |DCD_i(m) - DCD_i(0)|$ and m is the position with the lowest value in the DCD, N is the size of the DCD and i is the interval number.
3. SCD features: the improved DCD curve (1D) is approximated by a parabola. The first SCD feature is the quadratic coefficient of this parabola. The second is the “depth” of the parabola, defined as the height difference taken at dissolve segment boundaries and at the “center”.
4. VarProj: difference of the projection histograms extracted in the first step (cut detection).
5. Motion: motion vectors are also extracted in the first step, when the phase correlation method is computed, for each block we compute the magnitude of the motion vector.

3.7 Summary

The system that we propose in this step deals with a statistical learning approach for dissolve detection. Figure 4 shows the steps of our approach. The first step is the detection of possible dissolves, this step is based on three processes: the computation of illumination variance and the EAG, the extraction of candidate regions and, the verification of candidate regions. The second and third processes are executed for variance and for EAG. We filter these possible dissolves eliminating the intervals that corresponds to cuts and fades. We use our cut detection algorithm proposed in [30]. The third step consists of two processes: DCD confirmation and computation of DCD features. In the DCD confirmation we compute the DCD of each filtered possible interval and extract some characteristics of the DCD in the second process (quadratic coefficient and “deep” of the parabola).

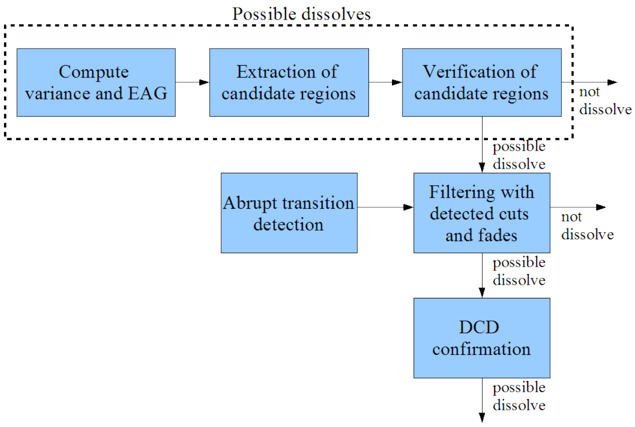


Figure 4: Proposed model for dissolve detection

4. FADE DETECTION

A fade process is a special case of a dissolve process. During a fade, a fade in a video sequence gradually darkens and is replaced by another image which either fades in or begins abruptly. Alattar [4] detects fades by recording all negative spikes in the second derivative of frame luminance variance curve. The drawback with this approach is that motion also would cause such spikes. Lienhart [31] proposes detecting fades by fitting a regression line on the frame standard deviation curve. Troung et. al. [6] observe the mean difference curve, examining *the constancy of its sign* within a potential fade region. We present further extensions to these techniques.

A fade-out process is characterized by a progressive darkening of a shot P until the last frame becomes completely black. A fade-in occurs when the picture gradually appears from a black screen. The fades can be used to separate different TV program elements such as the main show material from commercial blocks.

Fade-in and fade-out occur together as a fade group, i.e., a fade group starts with a shot fading out to a color $\mathscr{C}$ which is the followed by a sequence of monochrome frames of the same color, and it ends with a shot fading in from color $\mathscr{C}$.

As a fade is a special case of a dissolve we can explore some of the features used for dissolve detection. The salient features of our fade detection algorithm are the following:

1. The existence of monochrome frames is a very good clue for detecting all potential fades, these are used in our algorithm. In a quick fade, the monochrome sequence may be compound by a single frame while in a slower fade it would last up to 100 frames [6]. Therefore, detecting monochrome frames (candidate region) is the first step in out algorithm.
2. In this second step we are going to use a descriptor that characterize a dissolve, our improved DCD. The variance curves of fade-out and fade-in frame sequences have a half-parabolic shape independent of $\mathscr{C}$. Therefore, if we compute the DCD feature in the region where the fade-out occurs we will have a parabola shape, the same principle is applied for the fade-in.
3. We also constrain the variance of the starting frame of a fade-out and the ending of a fade-in to be above a threshold to eliminate false positives caused by dark scenes, thus preventing them from being considered as monochrome frames.

5. SUPPORT VECTOR MACHINE

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. 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.

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 evaluate similarities between data $K(d_t, d_s)$ in the input space.

We evaluate different kernel functions: linear, polynomial, gaussian radial basis, gaussian with χ^2 distance (Gauss-χ^2) and triangular [32]. Even though the result with the different kernels were almost the same, the Gauss-χ^2 was the best one. Thus, we use a gaussian with χ^2 distance $K(d_t, d_s) = e^{-\chi^2(d_t,d_s)/2\sigma^2}$ kernel function.

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. The choice of SVM as a classifier is due to the well known performance in statistical learning information retrieval.

6. EXPERIMENTATION

Our training set consists on a single video of 4197 frames (2mins. 20 secs.) with 50 cuts. This video is captured from a TV-station and is composed by a segment of commercials. We use a SVM classifier and train it with a gaussian with χ^2 distance kernel.

The nomenclature used for the features is as follows: RGB color histogram (Ch), HSV color histogram (HSVh), opponent color histogram (R-Gh), Zernike moments (Zer), Fourier-Mellin moments (Fou), Horizontal project histogram (Hor), Vertical projection histogram (Ver), Phase correlation (Ph) and Variance (Var). In Table 1, we present the visual feature vectors for cut detection used for the 10 runs.

In Table 1, we present the visual feature vectors for dissolve detection used for the 10 runs.

For fade detection we choose a threshold of 200 for the variance of each frame, if the variance is lower than that value we consider it as a monochrome frame and a possible fade. After that is necessary to see if the interval have two download parabolas, one for fade-in and other for fade-out.

In Table 2 we show the performance of our system for cut and gradual transition detection, measured in recall and precision.

Figure 5 shows the performance of our system for (a) all transitions and (b) for cut detection, (c) for gradual transition, measured in recall and precision and (d) measures the accuracy of gradual transitions. We compare our results to all other submissions.

We want to emphasize with these results that our system for cut detection is very robust to training data set. Indeed, the training data set used here is brazilian TV videos which are very different in terms of quality, format and length from TRECVID videos we used for testing our system.

7. CONCLUSIONS

We present our hierarchical system for shot boundary detection. The first step is dedicated to sharp cut detection using learning-based approach. Then we seek for gradual transition inside the shots delimited by the sharp cuts resulting from first step. The hierarchical structure of our system allows us to reduce to two modalities the identification of gradual transitions: fast motion or dissolve. We provide new features to caracterize dissolves. We combine these features with other classic ones that describe the characteristics of the dissolve segments.

Even though our system only detects dissolves and fades, in the case of gradual transitions, the performance of the system is satisfactory. Our next step is to improve the performance of our hierarchical system including more edition effects detection. We want now to extend our video segmentation system to a video content-based retrieval system adapting our content analysis system Retin with the results of the system proposed in this paper.

sysID	Cuts	Gradual Transitions
Etis1	Ph, HSVh, Zer, Hor, Var	Ddata, VarProj
Etis2	Ph, HSVh, Ver, Hor, Var	Ddata, Motion
Etis3	Ph, HSVh, Ch, Fou, Zer, Var	Ddata, DCD
Etis4	Ph, Ch, Zer, Ver, Hor, Var	Ddata, DCD, SCD
Etis5	Ph, R-Gh, HSVh, Ch, Fou, Hor, Var	Ddata, DCD, VarProj
Etis6	Ph, HSVh, Ch, Fou, Zer, Hor, Var	Ddata, DCD, Motion
Etis7	Ph, Ch, Fou, Zer, Ver, Hor, Var	Ddata, SCD
Etis8	Ph, HSVh, Zer, Ver, Hor, Var	Ddata, SCD, VarProj
Etis9	Ph, R-Gh, HSVh, Ch, Fou, Zer, Hor, Var	Ddata, SCD, Motion
Etis10	Ph, HSVh, Ch, Fou, Zer, Hor, Ver, Var	Ddata, SCD, VarProj, Motion

Table 1: 10 best combinations of visual features for cuts and gradual transitions

Run	All transitions		Cuts		Gradual Transitions			
	Recall	Precision	Recall	Precision	Recall	Precision	F-Recall	F-Precision
Etis1	0.757	0.876	0.821	0.909	0.585	0.771	0.766	0.849
Etis2	0.764	0.868	0.825	0.889	0.602	0.798	0.773	0.850
Etis3	0.768	0.888	0.818	0.908	0.632	0.825	0.775	0.849
Etis4	0.771	0.879	0.827	0.886	0.621	0.853	0.775	0.849
Etis5	0.771	0.851	0.832	0.876	0.607	0.769	0.769	0.847
Etis6	0.761	0.861	0.828	0.876	0.581	0.807	0.774	0.849
Etis7	0.769	0.878	0.827	0.886	0.612	0.849	0.775	0.850
Etis8	0.762	0.850	0.821	0.879	0.604	0.758	0.770	0.849
Etis9	0.751	0.894	0.813	0.911	0.586	0.837	0.772	0.851
Etis10	0.743	0.842	0.803	0.868	0.583	0.757	0.767	0.843

Table 2: Detailed results for all runs for various settings.

8. ACKNOWLEDGMENTS

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

REFERENCES

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

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

[3] R. Lienhart, C. Kuhmunch, and W. Effelsberg, "On the detection and recognition of television commercials," *ICMCS*, pp. 509 – 516, 1997.

[4] A.M. Alattar, "Detecting and compressing dissolve regions in video sequences with a dvi multimedia image compression algorithm," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, pp. 13–16, 1993.

[5] Alan Hanjalic, "Shot boundary detection: Unraveled and resolved?," *CSVT*, vol. 12, no. 2, pp. 90–105, 2002.

[6] Ba Tu Truong, Chitra Dorai, and Svetha Venkatesh, "New enhancements to cut, fade, and dissolve detection processes in video segmentation," in *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, 2000, pp. 219–227.

[7] H.J. Zhang, A. Kankanhalli, and S.W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, vol. 1, no. 1, pp. 10–28, 1993.

[8] R. Zarih, J. Miller, and M. Kai, "Feature-based algorithms for detecting and classifying scene breaks," in *ACM Int. Conf. on Multimedia*, San Francisco, November 1996, pp. 97–103.

[9] Seung-Hoon Han and In So Kweon, "Detecting cuts and dissolves through linear regression analysis," *Electronics Letters*, vol. 39, no. 22, pp. 1579–1581, 2003.

[10] P. Campisi, A. Neri, and L. Sorgi, "Automatic dissolve and fade detection for video sequences," in *14th International Conference on Digital Signal Processing, 2002. DSP 2002*, 2003, vol. 2, pp. 567–570.

[11] J.S. Boreczky and L.D. Wilcox, "A hidden markov model framework for video segmentation using audio and image features," in *ICASSP'98*, 1998, vol. 6, pp. 3741–3744.

[12] John Adcock, Andreas Gingensohn, Matthew Cooper, Ting Liu, Lynn Wilcox, and Eleanor Rieffel, "Fxpal experiments for trecvid 2004," in *TREC Video Retrieval Evaluation Online Proceedings: TRECVID 2004*, 2004.

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

[14] Y. Qi, T. Liu, and A. Hauptmann, "Supervised classification of video shot segmentation," in *ICME*, Baltimore, MD, July 6-9 2003.

[15] P.-H. Gosselin and M. Cord, "Precision-oriented active selection for interactive image retrieval," in *ICIP*, October 2006.

[16] Simon Tong, *Active Learning: Theory and Applications*, Ph.D. thesis, Stanford University, 2001.

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

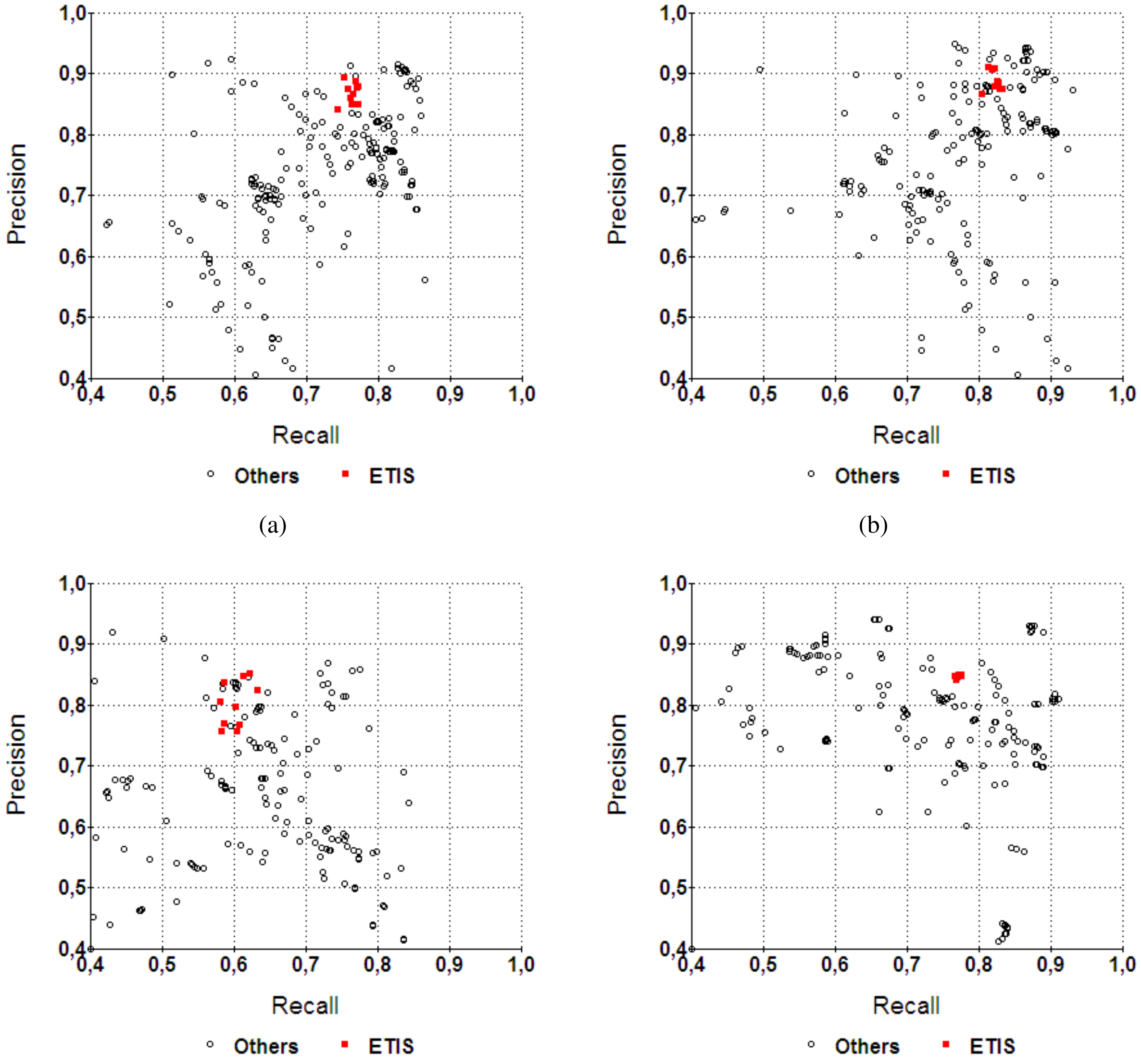
[18] P.H. Gosselin and M. Cord, "A comparison of active classification methods for content- based image retrieval," in *CVDB*, Paris, France, June 2004, pp. 51–58.

[19] C. Kan and M.D. 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.*, 2001, pp. 511–515.

[20] K. Whoi-Yul and K. Yong-Sung, "A region-based shape descriptor using zernike moments," *Image Communication*, vol. 16, no. 95-102, 2000.

[21] O.D. Trier, A.K. Jain, and T. Taxt, "Feature extraction methods for character recognition - a survey," *Pattern Recognition*, vol. 29, pp. 641–662, 1996.

[22] James Ze Wang, "Methodological review - wavelets and imaging informatics : A review of the literature," *Journal of Biomedical Informatics*, pp. 129–141, July 2001, Avaliable on `http://www.idealibrary.com`.



(a)

(b)

(c)

(d)

Figure 5: Precision/Recall measure of performance on the TRECVID 2006. (a) show our results for all transitions, (b) show our results for abrupt transitions (cuts), (c) for gradual transition and (d) for accuracy of gradual transitions, measured in frame-precision/frame-recall

[23] S. V. Porter, M. Mirmehdi, and B. T. Thomas, “Temporal video segmentation and classification of edit effects,” *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1097–1106, December 2003.

[24] Jing-Un Won, Yun-Su Chung, In-Soo Kim, Jae-Gark Choi, and Kil-Houm Park, “Correlation based video-dissolve detection,” in *International Conference on Information Technology: Research and Education, 2003*, 2003, pp. 104 – 107.

[25] J. Nam and A.H. Tewfik, “Detection of gradual transitions in video sequences using b-spline interpolation,” *IEEE Transactions on Multimedia*, vol. 7, pp. 667–679, 2005.

[26] Hong Heather Yu and Wayne Wolf, “A hierarchical multiresolution video shot transition detection scheme,” *Comput. Vis. Image Underst.*, vol. 75, no. 1-2, pp. 196–213, 1999.

[27] H. Yu, G. Bozdagi, and S. Harrington, “Feature-based hierarchical video segmentation,” in *ICIP*, 1997, vol. 2, pp. 498–501.

[28] Seung-Hoon Han and In So Kweon, “Detecting cuts and dissolves through linear regression analysis,” *Electronics Letters*, vol. 39, no. 22, pp. 1579–1581, October 2003.

[29] R. Lienhart, “Reliable transition detection in videos: A survey and practitioner’s guide.,” *IJIG*, vol. 1, no. 3, pp. 469 – 486, 2001.

[30] G. Cámara-Chávez, M. Cord, F. Precioso, S. Philipp-Foliguet, and Arnaldo de A. Araújo, “Robust scene cut detection by supervised learning,” in *EUSIPCO*, 2006.

[31] R. Lienhart, “Comparison of automatic shot boundary detection algorithms,” in *Proc. SPIE. Storage and Retrieval for Image and Video Databases VII*, December 1998, vol. 3656, pp. 290–301.

[32] G. Cámara-Chávez, M. Cord, F. Precioso, S. Philipp-Foliguet, and Arnaldo de A. Araújo, “Video segmentation by supervised learning,” in *19th Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI 2006*, 2006.