The COST292 experimental framework for TRECVID 2007

Q. Zhang,[1] M. Corvaglia,[2] S. Aksoy,[3] U. Naci,[4]
N. Adami,[2] N. Aginako,[12] A. Alatan,[7] L. A. Alexandre[10], P. Almeida[10], Y. Avrithis,[8]
J. Benois-Pineau,[6] K. Chandramouli,[1] U. Damnjanovic,[1] E. Esen,[7] J. Goya,[12]
M. Grzegorzek,[1] A. Hanjalic,[4] E. Izquierdo,[1] R. Jarina[11], P. Kapsalas,[8]
I. Kompatsiaris,[5] M. Kuba[11], R. Leonardi,[2] L. Makris,[5] B. Mansencal,[6] V. Mezaris,[5]
A. Moumtzidou,[5] P. Mylonas,[8] S. Nikolopoulos,[5] T. Piatrik,[1] A. M. G. Pinheiro[10],
B. Reljin,[9] E. Spyrou,[8] G. Tolias,[8] S. Vrochidis,[5] G. Yakın,[3] G. Zajic[9]

February 26, 2008

Abstract

In this paper, we give an overview of the four tasks submitted to TRECVID 2007 by COST292. In shot boundary (SB) detection task, four SB detectors have been developed and the results are merged using two merging algorithms. The framework developed for the high-level feature extraction task comprises four systems. The first system transforms a set of low-level descriptors into the semantic space using Latent Semantic Analysis and utilises neural networks for feature detection. The second system uses a Bayesian classifier trained with a "bag of subregions". The third system uses a multi-modal classifier based on SVMs and several descriptors. The fourth system uses two image classifiers based on ant colony optimisation and particle swarm optimisation respectively. The system submitted to the search task is an interactive retrieval application combining retrieval functionalities in various modalities with a user interface supporting automatic and interactive search over all queries submitted. Finally, the rushes task submission is based on a video summarisation and browsing system comprising two different interest curve algorithms and three features.

[1] Q. Zhang, K. Chandramouli, U. Damnjanovic, T. Piatrik and E. Izquierdo are with Dept. of Electronic Engineering, Queen Mary, University of London, Mile End Road, London E1 4NS, UK, {qianni.zhang,uros.damnjanovic,tomas.piatrik,krishna.chandramouli,ebroul.izquierdo}@elec.qmul.ac.uk

[2] M. Corvaglia, N. Adami and R. Leonardi are with University of Brescia, Via Branze 38 25123 Brescia, ITALY, {marzia.corvaglia,nicola.adami,riccardo.leonardi}@ing.unibs.it

[3] G. Yakın and S. Aksoy are with RETINA Vision and Learning Group, Bilkent University, Bilkent, 06800, Ankara, Turkey, {gyakin@ug,saksoy@cs}.bilkent.edu.tr

[4] U. Naci, A. Hanjalic are with Delft University of Technology, Mekelweg 4, 2628CD, Delft, The Netherlands, {s.u.naci,A.Hanjalic}@tudelft.nl

[5] S. Vrochidis, A. Moumtzidou, S. Nikolopoulos, V. Mezaris, L. Makris and I. Kompatsiaris are with Informatics and Telematics Institute/Centre for Research and Technology Hellas, 1st Km. Thermi-Panorama Road, P.O. Box 361, 57001 Thermi-Thessaloniki, Greece, {stefanos, moumtzid, nikolopo, bmezaris, lmak, ikom}@iti.gr

[6] B. Mansencal and J. Benois-Pineau are with LABRI, University Bordeaux, 351, cours de la Liberation 33405, Talence, France, jenny.benois,boris.mansencal@labri.fr

[7] E. Esen, A. Alatan are with Middle East Technical University, 06531, Ankara, Turkey, alatan@eee.metu.edu.tr, ersin.esen@bilten.metu.edu.tr

[8] E. Spyrou, P. Kapsalas, G. Tolias, P. Mylonas and Y. Avrithis are with Image Video and Multimedia Laboratory, National Technical University of Athens, Athens, Greece, {espyrou,pkaps,gtolias,fmylonas,iavr}@image.ntua.gr

[9] B. Reljin and G. Zajic are with Faculty of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11000 Belgrade, Serbia, reljinb@etf.bg.ac.yu

[10] A. M. G. Pinheiro, L. A. Alexandre and P. Almeida are with the Universidade da Beira Interior, Covilha, Portugal, pinheiro@ubi.pt, {lfbaa,palmeida}@di.ubi.pt

[11] R. Jarina and M. Kuba are with Department of Telecommunications, University of Zilina. Univerzitna 1, 010 26 Zilina, Slovakia, {jarina,kuba}@fel.uniza.sk

[12] N. Aginako, J. Goya are with VICOMTech, Mikeletegi Pasealekua, 57 Parque Tecnolgico 20009 Donostia / San Sebastin, Spain,{naginako,jgoya}@vicomtech.es

1 Introduction

This paper describes collaborative work of several European institutions in the area of video retrieval under a research network COST292. COST is an intergovernmental network which is scientifically completely self-sufficient with nine scientific COST Domain Committees formed by some of the most outstanding scientists of the European scientific community. Our specific action COST292 on semantic multi-modal analysis of digital media falls under the domain of Information and Communication Technologies.

Being one of the major evaluation activities in the area, TRECVID has always been a target initiative for all COST292 participants [1]. Therefore, this year our group has submitted results to all four tasks. Based on our submissions to TRECVID last year, we have tried to improve and enrich our algorithms and systems according the previous experience [2]. The following sections bring details of applied algorithms and their evaluation.

2 Shot Boundary Detection Task

In shot boundary (SB) detection task, four SB detection algorithms have been developed, by the University of Brescia (U. Brescia), the Technical University of Delft (TU Delft), the Middle East Technical University (METU) and the Queen Mary, Univeristy of London (QMUL) respectively. These algorithms are applied on TRECVID 2007 audiovisual contents and the results are merged using two algorithms provided by the LaBRI, University of Bordeaux 1 (LaBRI) and U. Brescia, in order to investigate how and how much the performance can be improved.

In the following sections, first the tools proposed by each COST292 participants are presented, then the integration methods are described, finally the results of submission are reported.

2.1 SB Detector of University of Brescia

The algorithm[13] is based on the classical twin comparison method where the error signals used to detect transitions is based on statistical modelling. Assuming that the contents of two consecutive shots can be represented by two independent random variables, an abrupt transition is modelled as drastic variation in the colour density function of adjacent frames while dissolves are detected evaluating the difference between the colour density function of the actual frame and the one predicted by the dissolve model (Figure 1). Adaptive thresholds are used to improve the performance [3].

The detection of **gradual transition** is limited only to dissolves and fades using the algorithm described in [4]. In this model, two basic assumptions are considered. The first requires that series of video frames forming a given shot can be modelled by a stationary process, at least for a duration equal to the dissolve length. If this hypothesis is satisfied, a single frame can be used to describe the statistics of at least a clip of video frames. The second assumption implies the independence between the RVs describing the frames of adjacent shots. An estimate of the marginal pdf of each process is represented by the last frame of $shot_{out}/shot_{in}$ prior to the dissolve, F_{in}/F_{out}, respectively.

If these two assumption are satisfied, the colour or luminance histogram of a frame belonging to dissolve can be obtained as convolution of the histograms H_{in} and H_{out} properly scaled to take into account the dissolve frame weighting. This implies that the difference between $H[n]$ and $H_{in} * H_{out}$ should ideally be zero. On the contrary, if F_{in} and F_{out} belong to a same shot, the previous histogram difference would be non-zero. From this consideration, it is possible to obtain a simple criterion for dissolve detection.

In **hard transition** detection, in order to reduce the influence of motion, the video frames are partitioned into a grid of rectangles and for each area the colour histogram is extracted. The distance between histograms extracted from consecutive frames are then calculated. For transition detection, initially a series on M frames is loaded in Buffer and for each of them the distance introduced above are evaluated. All this information are then used to adaptively estimate the thresholds used in the twin comparison detection scheme. Once the detection process is started, for each frame of the video sequence, the probability belonging to a hard and a gradual transition are evaluated and the frames are classified accordingly. The confidence values are provided directly as detection probability.

[13] The proposed method has been developed in collaboration with TELECOM Italia.

2.2 SB Detector of the TU Delft

The proposed method introduces the concept of *spatiotemporal block based analysis* for the extraction of low level events. The system makes use of the overlapping 3D pixel blocks in the video data as opposed to the many other methods that use the frames or the 2D blocks in the frames as the main processing units. The full detail of the system can be found in [5].

The method is based on the gradient of spatiotemporal pixel blocks in video data. Derivatives in the temporal direction $\vec{\mathbf{k}}$ and the estimated motion direction $\vec{\mathbf{v}}$ are extracted from each data block (i, j, k) of size C_x, C_y and C_t as in the following equation.

$$\nabla_{\vec{\mathbf{v}}} I_{i,j,k}(m, n, f) = I_{i,j,k}(m + v_x, n + v_y, f + 1) - I_{i,j,k}(m, n, f) \tag{1}$$

where I is the pixel intensity function and $\vec{\mathbf{v}} = (v_x, v_y)$, iis the estimated motion direction. We also calculate $\nabla_{\vec{\mathbf{k}}} I_{i,j,k}(m, n, f)$ where $\vec{\mathbf{k}} = (0, 0)$, assuming zero motion. We calculate two different measures from this derivative information, namely *the absolute cumulative luminance change*:

$$\nabla^a_{\vec{\mathbf{v}}} I_{i,j,k} = \frac{1}{C_x \cdot C_y} \sum_{m=0}^{C_x-1} \sum_{n=0}^{C_y-1} \sum_{f=0}^{C_t-2} |\nabla_{\vec{\mathbf{v}}} I_{i,j,k}(m, n, f)| \tag{2}$$

and *the average luminance change*:

$$\nabla^d_{\vec{\mathbf{v}}} I_{i,j,k} = \frac{1}{C_x \cdot C_y} \sum_{m=0}^{C_x-1} \sum_{n=0}^{C_y-1} \sum_{f=0}^{C_t-2} (\nabla_{\vec{\mathbf{v}}} I_{i,j,k}(m, n, f)) \tag{3}$$

Besides calculating the values (2) and (3), we keep track of the maximum time derivative value in a block. For each spatial location (m, n) in the block (i, j, k), we search for the frame $f^{max}_{i,j,k}(m, n)$, where the maximum luminance change takes place:

$$f^{max}_{i,j,k}(m, n) = \mathbf{argmax}(|\nabla_{\vec{\mathbf{v}}} I_{i,j,k}(m, n, f)|) \tag{4}$$

After the frames (4) are determined for each pair (m, n), we average the maximum time derivative values found at these frames for all pairs (m, n), that is

$$\nabla^{max}_{\vec{\mathbf{v}}} I_{i,j,k} = \frac{1}{C_x \cdot C_y} \sum_{m=0}^{C_x-1} \sum_{n=0}^{C_y-1} |\nabla_{\vec{\mathbf{v}}} I_{i,j,k}(m, n, f^{max}_{i,j,k}(m, n))| \tag{5}$$

For the detection of gradual changes two features are calculated using (2), (3) and(5):

$$F_1(i, j, k) = \mathbf{max}(|\nabla^d_{\vec{\mathbf{k}}} I_{i,j,k} / \nabla^a_{\vec{\mathbf{k}}} I_{i,j,k}|, |\nabla^{max}_{\vec{\mathbf{v}}} I_{i,j,k} / \nabla^a_{\vec{\mathbf{v}}} I_{i,j,k}|) \tag{6}$$

$$F_2(i, j, k) = 1 - \mathbf{min}(|\nabla^{max}_{\vec{\mathbf{k}}} I_{i,j,k} / \nabla^a_{\vec{\mathbf{k}}} I_{i,j,k}|, |\nabla^{max}_{\vec{\mathbf{v}}} I_{i,j,k} / \nabla^a_{\vec{\mathbf{v}}} I_{i,j,k}|) \tag{7}$$

The value of $F_1(i, j, k)$ equals to 1 if the function $I_{i,j,k}(m, n, f)$ is monotonous and gets closer to zero as the fluctuations in the function values increase. The higher the value of $F_2(i, j, k)$ (i.e. close to 1), the more gradual (smooth) are the variations in the function $I_{i,j,k}(m, n, f)$ over time. The confidence value for the existence of a gradual transition at any temporal interval $k = K$ is calculated by averaging the $F_1(i, j, K) \cdot F_1(i, j, K)$ over all spatial indices (i, j) at the corresponding interval K.

Detection of cuts and wipes are based on the values calculated in (4). To do this, all $f^{max}_{i,j,k}(m, n)$ values are fit to a plane equation and the error is calculated. Lower error values suggests an abrupt change in the corresponding block. If the plane approximation error values are low in all blocks and the same time index, we detect a ”cut”. On the other hand if the time indices for the planes are distributed in a short time interval, this suggests a “wipe”.

The matrix in Figure 2 depicts the confidence values for an eight-minute sports video that contains two cuts and two wipes. Each column depicts the values of confidences collected row by row from all blocks sharing the same time index k. The brightness level of matrix elements directly reveals the values of confidence. We observe that in case of a cut, high values of this feature are time-aligned, that is, they form a plane vertical to the time axis. On the other hand, a wipe is characterized by high feature values, which are not time-aligned, but distributed over a limited time interval.

2.3 SB Detector of the METU

Shots are classified into three groups as hard cuts, short gradual cuts consisting of three frames and long gradual cuts. Each group is handled differently. The operations are performed on the DC image of the luminance channel of the video frames. The DC image is preferred due to its robustness against small changes that do not correspond to shot boundaries as well as its contribution to computation time (depending on the video content 3-5% of real-time on an Intel Core2 Duo 2.0 GHz system).

For hard cuts Edge Difference (ED) and Pixel Difference (PD) values are utilised [6]. ED is computed for consecutive frames by counting the differences between the edge images, which are obtained by Sobel operator, and normalising the total difference to the number of pixels. PD is the normalised total absolute pixel difference of consecutive frames. ED and PD values of the whole video are fed to 2-Means clustering. Initial cluster means are experimentally chosen as ($ED = 0, PD = 0$) for non-hard cut and ($ED = 0.18, PD = 0.1$) for hard cut.

ED and PD are also used for short gradual cuts with the exception that for each frame they are also computed using the second preceding frame in addition to the previous one, which are denoted by index values one and two. At each frame short gradual cuts are detected by thresholding. For a short gradual cut, $\exists PD_i > \tau_1$, $\Sigma PD_i > \tau_2$, and $\forall ED_i > \tau_3$, where τ_1, τ_2, and τ_3 are experimentally chosen as 10, 5, and 0.2, respectively.

Long gradual cuts are found using the Edge Energy(EE), which is computed using the edge image of the frame. Similar to [6] U-curves are searched based on the assumption that in case of a gradual cut EE starts to decrease and attains its local minimum at the center of the gradual transition. The U-curves are located by calculating the Least Squares estimates of the slopes of left(m_L) and right(m_R) lines at each center frame using previous and next 6 frames. If $m_L < 0$, $m_R > 0$, and $m_R - m_L > 1$, the center of the candidate gradual cut is located. The start and end frames of the transition are determined by searching the frames where slopes diminishes. If the candidate is already found as a hard cut or short gradual cut, it is discarded. Then false-positives are first eliminated by analyzing normalized Histogram (with 16 bins) Difference(HD_{SE}) and Edge Difference(ED_{SE}) between start and end frames. For a correct gradual cut, $HD_{SE} > \tau_4$ and $ED_{SE} > \tau_5$, where τ_4 and τ_5 are experimentally chosen as 0.2 and 0.1, respectively. Secondly, if motion vectors are available in the compressed bitstream, motion analysis at the center frame is performed for further elimination. Let M_1 denote the total number of non-zero motion vectors, M_X denote the sum of x components of the motion vectors, and M_Y denote the sum of y components of the motion vectors. The motion parameters are computed as the average of the three consecutive frames around the center. If $M_1 < \tau_6$ and $(|M_X| + |M_Y|/2) < \tau_7$, where τ_6 and τ_7 are experimentally chosen as 150 and 350, respectively, for QCIF dimension, then the candidate is settled as a long gradual cut. However, during TRECVID 2007 tests, the motion vectors of the bitstream are discarded and not utilized during the final analysis.

2.4 SB Detector of the QMUL

Conventional shot detection methods analyse consecutive frame similarities. Therefore, most of the general information about the shot is lost. Our work is motivated by the assumption that a shot boundary is a global feature of the shot rather than local. General knowledge about the shot is cumulated during the time from the information included in every frame. Information about the boundary is extracted indirectly from the information about the interaction between two consecutive shots. Spectral clustering aggregates contribution of each frame in the form of the objective function. By optimising the objective function when clustering frames, individual shots are separated, and cluster bounds are used for detecting shot boundaries. By keeping the information in the objective function, every frame of the shot has its contribution to the overall information about the boundary. As frames that belong to the same shot are temporally aligned, cluster borders will be points on the time axis. In our algorithm, spectral clustering algorithm Normalised Cut [7] is used for clustering. Clustering is performed inside the sliding window, until the whole video is analysed. Firstly, a similarity matrix is created by applying a specific similarity measure over the set of features. Three eigenvectors corresponding to the second, third and fourth smallest egienvalues are used for describing the structure of video inside the sliding window. Every eigenvector indicates possible choice of the shot boundary. Specially created heuristics is then applied to analyse results of the clustering and give final results.

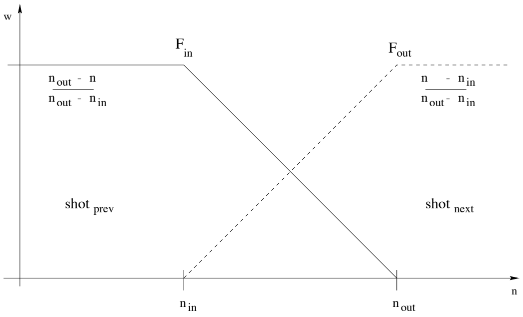

Figure 1: A possible frames weighting for dissolves (U. Brescia).

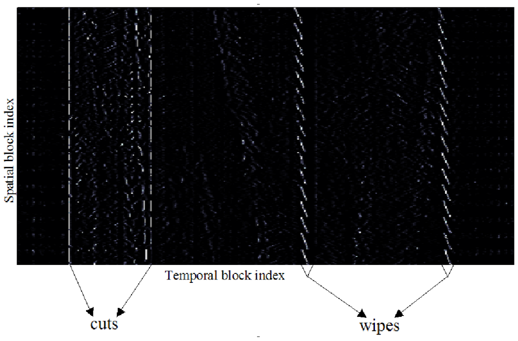

Figure 2: An illustration of confidence values for block based abrupt changes (TU Delft).

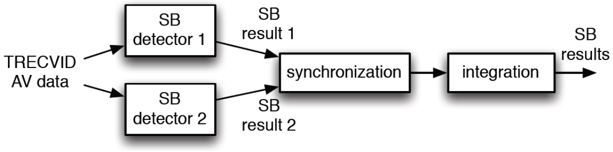

Figure 3: Merging framework for SBD result integration.

2.5 Merging

The merging method proposed in [2] is based on the knowledge of DB detectors in terms of performance. Such method also assumes that the confidence values are reliable.

This year, this algorithm has been improved with the purpose of proposing a general method which performs a kind of *blind integration*. In other words, we supposed to do not know the DB detectors but to only know the SB detector results. The new merging method is based on the framework shown in Figure 3. The results of each SB detector can be combined with the results of one of the other three SBD detectors. The integration process requires a synchronisation process because each DB detector uses its own decoder. If the confidence values are available and reliable, the integration is performed as follow. Let $B_1 = b_1$ the set of transitions of SB detector 1 and $B_2 = b_2$ the set transition of SB detector 2, c_1 and c_2 the associated confidence measures, and C_1 and C_2 two thresholds with $C_1 < C_2$. If a SB $b_1 \in B_1$ does not intersect any SB $b_2 \in B_2$, and if $c_1 > C_1$, then b_1 is retained as a detection result. If a SB $b_2 \in B_2$ does not intersect any SB $b_1 \in B_1$, and if $c_2 > C_2$, then b_2 is retained as a detection result. In the case of $b_1 \cap b_2$, b_1 is retained as a detection result. While, if the confidence value is not available or reliable, two possible integrations can be generated: all transitions available in both B_1 and B_2 or all the transitions given by the intersection of B_1 and B_2.

2.6 Results

COST 292 submitted ten runs. Four runs have been submitted individually from each COST292 participants. The overall recall and precision of these runs are respectively: 0.871 and 0.762 for U. Brescia, 0.905 and 0.650 for METU, 0.727 and 0.531 for QMUL, 0.802 and 0.578 for TU Delft.

The remaining 6 submitted runs have been obtained with the merging algorithms described above. The optimum couple of DB detectors for integration has been chosen on the training performed on TRECVID 2006 data. Since the integration was blind, two runs failed while four runs were successful. Among the four successful runs, three have been obtained using the confidence value (SB_{1c}, SB_{2c}, SB_{3c}) while one ignoring the confidence value because it was not reliable (SB_{1min}). The overall recall and precision obtained are respectively: 0.795 and 0.607 for SB_{1c}, 0.877 and 0.792 for SB_{2c}, 0.877 and 0.792 for SB_{3c}, 0.756 and 0.786 SB_{1min}. We can note that the individual runs can be improved by the proposed merging method.

3 High-level feature extraction

COST292 participated to the high-level feature extraction task with four separate systems as well as with integrated runs that combine these systems. The first system, developed by the National Technical University of Athens (NTUA), transforms a set of low-level descriptors into the semantic space using Latent Semantic Analysis and utilises neural networks for feature detection and is described in Section 3.1. The second system, developed by the Bilkent University (Bilkent U.), uses a Bayesian classifier trained with a "bag of subregions" and is described in Section 3.2. The third system, by the University of Beira Interior (UBI), uses a multi-modal classifier based on SVMs and several descriptors and is described in Section 3.3. The fourth system, by QMUL, uses two image classifiers based on ant colony optimisation and particle swarm optimisation respectively, and is described in Section 3.4.

3.1 Feature extractor from NTUA

In the NTUA system, for the detection of all concepts apart from *person* and *face*, we use the following approach [8]. We choose to extract colour and texture MPEG-7 descriptors from the keyframes and more specifically *Dominant Color*, *Scalable Color*, *Color Layout*, *Homogeneous Texture* and *Edge Histogram*. These low-level descriptions are extracted from image regions that resulted from a coarse colour segmentation. Then, a clustering algorithm is applied on a subset of the training set, in order to select a small set of regions that will be used to represent the images. From each cluster we select the closest region to the centroid. This region will be referred to as "region type". Then, for each keyframe we form a model vector description. Let: $d_i^1, d_i^2, ..., d_i^j, i = 1, 2, \ldots, N_R$ and $j = N_C$, where N_C denotes the number of region types, N_R the number of the regions within the image and d_i^j is the distance of the i-th region of the image to the j-th region type. The model vector D_m is formed as

$$D_m = \left[min\{d_i^1\}, min\{d_i^2\}, ..., min\{d_i^{N_C}\} \right], i = 1, 2, \ldots, N_R. \tag{8}$$

Then we follow a Latent Semantic Analysis approach. We construct the co-occurrence matrix of region types in given keyframes of the training set. After the construction of the co-occurrence matrix, we solve the SVD problem and transform all the model vectors to the semantic space. For each semantic concept, a separate neural network (NN) is trained. Its input is the model vector in the semantic space and its output represents the confidence that the concept exists within the keyframe. This model is applied to the TRECVID video sequences to construct detectors for the following features: *desert*, *vegetation*, *mountain*, *road*, *sky* ,*fire-explosion*, *snow*, *office*, *outdoor*, *face* and *person*.

The main idea of our *face* and *person* detection algorithm is based on extracting regions of interest, grouping them according to some similarity and spatial proximity predicates and subsequently defining whether the area obtained, represents a human body. Thus, the method initially involves detection of salient points and extraction of a number of features representing local the colour and texture. At the next step, the points of interest are grouped with the aid of an unsupervised clustering algorithm (DBSCAN) that considers the density of the feature points to form clusters. In the classification stage, there is a need for a robust feature set allowing the human form to be discriminated even in a cluttered background. Histogram of Oriented Gradients (HoG) descriptor [9] is used to encode information associated to the human body boundary. The method is based on evaluating well-normalised local histograms of image gradient orientations in a dense grid. The basic idea is that local object appearance and shape can often be characterised rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions. In practice, this is implemented by dividing the image window into small spatial regions ("cells"), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. For better invariance to illumination, shadowing, etc., it is also helpful to perform contrast normalisation to the local responses before using them. Finally our human detection chain involves tiling the detection window with a dense grid of HoG descriptors and using the feature vector in a conventional SVM based window classifier.

3.2 Feature extractor from Bilkent University

The detectors developed by Bilkent U. exploit both colour and spatial information using a bag-of-regions representation [10].The first step is the partitioning of keyframes into regions. After experimenting with several segmentation algorithms, we decided to use the k-means with connectivity constraint algorithm [11]. After an image is segmented into several regions, each region is modelled using the multivariate histogram of the HSV values of its pixels with 8 bins used for the H channel and 3 bins for each of S and V channels, resulting in a 72-dimensional feature vector. Then, a codebook of region types is constructed using the k-means algorithm for vector quantisation. The number of codewords (k) was set to 1000 empirically. The output of this step is a discrete type label assigned to each region.

Colour information can be very useful in discriminating objects/regions in an image if they have very distinct colours. However, just like any other low-level features, colour cannot distinguish conceptually different objects/regions if they fall to nearby locations in the feature space. An important element of image understanding is the spatial information. For example, finding a region with dominant blue colour (that may be water) and a neighbouring beige region (that may be sand) with another blue region (that may be sky) above them can increase the possibility of being a coast image. Furthermore, two images with similar regions but in different spatial locations can have different interpretations. Hence, spatial information can be used to resolve ambiguities in image classification.

Different methods have been proposed to model region spatial relationships. However, it becomes a combinatorial problem if one tries to model all possible relationships between regions in an image. Therefore, we decided to use only the vertical relationship of "above-below" because it arguably provides a better characterisation of the content. For example, flipping a photograph horizontally does not usually alter its semantics but flipping it vertically or rotating it greatly perturb its perception. To determine the vertical relative position of two regions, we use their projections on both axes. If there is an overlap between the projections on the x-axis, their projections on the y-axis are compared. If they have no overlap on the y-axis or if the overlap is less than 50 percent of the area of the smaller region, we conclude that the one with a greater centroid ordinate is above the other one. If these overlap criteria are not met, it is concluded that no significant vertical relative arrangement exists between these two regions. The result of this step is a list of region pairs that satisfy the "above-below" relationship for each image.

After each region is assigned a type label and the pairwise spatial relationships are computed, each image is represented as a "bag-of-regions". We consider two settings for this bag-of-regions representation: 1) each region is regarded separately and a "bag of individual regions" representation is generated, and 2) regions that satisfy the above-below relationship are grouped together and a "bag of region pairs" representation is constructed. Finally, these two representations are used separately to train Bayesian classifiers. Given the positive examples for each semantic concept (high-level feature), using multinomial density models, the probability values needed by the Bayesian decision rule are computed using the maximum likelihood estimates.

3.3 Feature extractor from UBI

UBI approach to detect high-level features is based on the detection of several descriptors that are used for a multi-modal classification after a suitable training.

The main descriptor is the HoGs such as the ones proposed in [9]. The number of directions considered depended on the type of object that was to be detected: either 9 or 18 directions. This description information was complemented with colour information from the RGB colour histograms, texture information from 9-7 bi-orthogonal filters, colour correlograms for 1 pixel distance with a colour quantisation to 16 colours, dominant colour descriptor, a combination of shape and texture information using a Scale-Space Edge Pixel Directions Histogram.

The keyframes were subdivided into rectangular sub-images of varied size, depending on the descriptor. These sub-images were processed individually and a classification was obtained from an SVM with RBF kernel. The result for a shot was obtained by averaging the classification scores of each of its sub-images in the keyframe.

3.4 Feature extractor from QMUL

The system developed by QMUL uses two image classifiers: classifier based on ant colony optimisation (ACO) and classifier based on particle swarm optimisation (PSO).

The idea underpinning the ACO model is loosely inspired by the behavior of real ants. The real power of ants resides in their colony brain and pheromone-driven communication within the colony. An important and interesting behavior of ant colonies is, in particular, how ants can find the shortest paths between food

Table 1: Numerical results for high-level feature extraction.

	QMUL	NTUA	UBI	Bilkent U.	Sum	Product
Total true shots returned	57	56	134	646	683	629
Mean (inferred average precision)	0.002	0.001	0.004	0.010	0.014	0.011

sources and their nest. For image classification task, the ACO algorithm is implemented and it is integrated with the semi-supervised COP-K-means approach.

In our proposal, the ACO plays its part in assigning each image to a cluster and each ant is giving its own classification solution. Images are classified based on the probability influenced by heuristic information and pheromone value. The main idea of finding optimal solution resides in marking classification solutions by pheromone as follows:

$$\tau_{(X_i,C_j)}(t) = \rho\, \tau_{(X_i,C_j)}(t-1) + \sum_{a=1}^{m} \Delta\tau^{a}_{(X_i,C_j)}(t) \tag{9}$$

where ρ is the pheromone trail evaporation coefficient $(0 \leq \rho \leq 1)$ which causes vanishing of the pheromones over the iterations. $\tau_{(X_i,C_j)}(t-1)$ represents the pheromone value from previous iteration. $\Delta\tau^{a}_{(X_i,C_j)}(t)$ is a new amount of pheromones calculated from all m ants that assign image X_i to the j'th cluster. Definition of $\Delta\tau^{a}_{(X_i,C_j)}(t)$ ensure that the pheromone increases when clusters get more apart and when each cluster has more similar images. The ACO makes the COP-K-means algorithm less dependent on the initial parameters and distribution of the data; hence it makes it more stable. Furthermore the ACO based multi-modal feature mapping improves inferring semantic information from low-level feature.

PSO technique is one of the meta-heuristic algorithms inspired by Biological systems. The image classification is performed using the self organising feature map (SOFM) and optimising the weight of the neurons by PSO [12]. The algorithm is applied to SOFM for optimising the weights of the neurons. The objective of SOFM is to represent high-dimensional input patterns with prototype vectors that can be visualised in a usually two-dimensional lattice structure [13]. Input patterns are fully connected to all neurons via adaptable weights, and during the training process, neighbouring input patterns are projected into the lattice, corresponding to adjacent neurons. SOFM enjoys the merit of input space density approximation and independence of the order of input patterns. Each neuron represents an image with dimension equal to the feature vector. Two different SOFM networks were used in detecting features. The first network configuration is a dual layer SOFM (DL-SOFM) structure which enables training of only positive models while the negative training models are implicitly generated by the network property. This model provides a high degree of recall, while the second configuration is a single layer rectangular mesh (R-SOFM), enabling explicit training of both positive and negative models. Thus enabling to achieve high precision.

3.5 Results

In addition to individual runs where the output from each system is submitted separately (QMUL: COST292R1, NTUA: COST292R2, UBI: COST292R3, Bilkent U.: COST292R4), we combined the confidence values for each shot for each high-level feature using the sum and product combination rules [14] and generated the runs COST292R5 and COST292R6, respectively. The numerical results are shown in Table 1. Among the 20 features evaluated by NIST, our detectors had relatively better success (among all submitted runs by COST292) for the following concepts: sports, office, meeting, mountain, waterscape, animal, screen, airplane, car, and boat/ship.

4 Interactive Search

The system submitted to the search task is an interactive retrieval application developed jointly by the Informatics and Telematics Institute(ITI-CERTH), QMUL, University of Zilina (U. Zilina) and University of Belgrade (U. Belgrade). It combines basic retrieval functionalities in various modalities (i.e. visual, audio, textual) and a user friendly graphical interface, as shown in Figure 4, that supports the submission of queries using any combination of the available retrieval tools and the accumulation of relevant retrieval

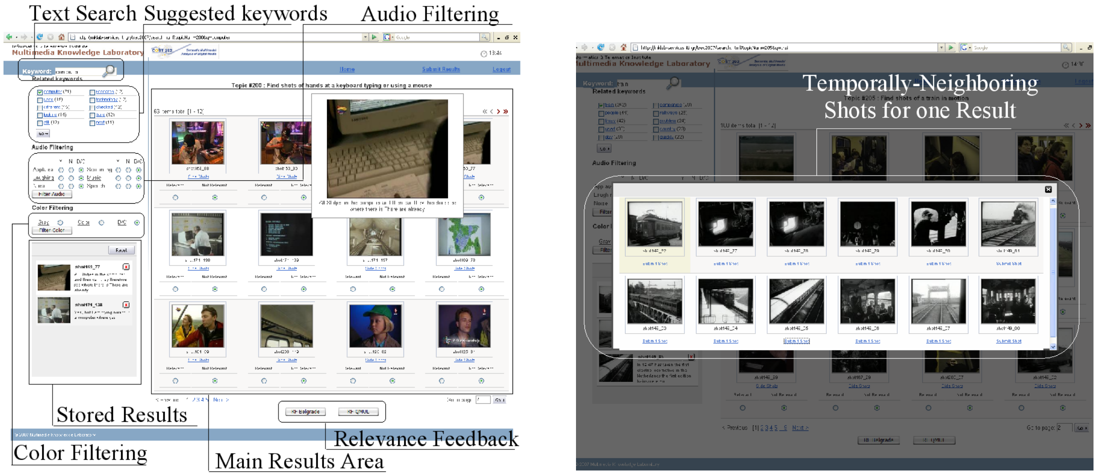

Figure 4: User interface of the interactive search platform

results over all queries submitted by a single user during a specified time interval. The following basic retrieval modules are integrated in the developed search application:

- Visual Similarity Search Module;
- Audio Filtering Module;
- Textual Information Processing Module;
- Two different Relevance Feedback Modules.

The search system, combining the aforementioned modules, is built on web technologies, and more specifically php, JavaScript and a mySQL database, providing a GUI for performing retrieval tasks over the internet. Using this GUI, the user is allowed to employ any of the supported retrieval functionalities and subsequently filter the derived results using audio and colour constraints. The retrieval results (representative keyframes of the corresponding shots) are presented ordered by rank in descending order, providing also links to the temporally neighbouring shots of each one. The identities of the desirable shots, which are considered as relevant to the query, can be stored by the user (Figure 4). The latter is made possible using a storage structure that mimics the functionality of the shopping cart found in electronic commerce sites and is always visible through the GUI. In this way, the user is capable of repeating the search using different queries each time (e.g. different combination of the retrieval functionalities, different keywords, different images for visual similarity search, etc.), without loosing relevant shots retrieved during previous queries submitted by the same user during the allowed time interval. This interval is set to 15 minutes for the conducted experiments, in accordance with TRECVID guidelines. A detailed description of each retrieval module employed by the system is presented in the following section.

4.1 Retrieval Module Description

4.1.1 Visual similarity search

In the developed application, content based similarity search is realised using MPEG-7 visual descriptors capturing different aspects of human perception such as colour and texture. Specifically, five MPEG-7 descriptors namely Color Layout, Color Structure, Scalable Color, Edge Histogram, Homogeneous Texture are extracted from each image of the collection are extracted [15] and stored in a relational database. By concatenating these descriptors a feature vector is formulated to compactly represent each image in the multidimensional space. An r-tree structure is constructed off-line by using the feature vectors of all images and the corresponding image identifiers. R-tree(s) [16] are structures suitable for indexing multidimensional objects and known to facilitate fast and efficient retrieval on large scale. Principal Component Analysis (PCA) was also employed to reduce the dimensionality of the initial space. In the query phase, a feature vector is extracted from the example image and submitted to the index structure. The set of resulting numbers correspond to the identifiers of the images that are found to resemble the query one. Since the order of these identifiers is not ranked according to their level of similarity with the query example, an additional step for ranking these images using custom distance metrics between their feature vectors is further applied to yield the final retrieval outcome.

4.1.2 Textual information processing module

The textual query module attempts to exploit the shot audio information in the best way. This audio information is processed off-line with the application of Automatic Speech Recognition and Machine Translation to the initial video, so that specific sets of keywords can be assigned to each shot. The text algorithm employed by the module is the BM25 algorithm, which incorporates both normalised document length (the associated text for every image/key-frame, in our case) and term frequency. Appropriate values for the parameters used by BM25 have been selected as reported in [17] to produce satisfactory results. Enhancing this approach, the module is further capable of providing related keywords to the searcher by processing the associated text of the initial results and eventually extracting the most frequent keywords. In that way the module receives feedback from the results and suggests additional input to the user for submitting similar queries. Although the performance of the module is satisfactory in terms of time-efficiency, the quality of the results greatly depends on the reliability of the speech transcripts.

4.1.3 Audio filtering tool from University of Zilina

A user has an option to use additional filtering of the search results by applying audio content based filtering on the shots retrieved. Should a certain sound occurs in the video shots, a user has an option either to take or omit such shots from the list of shots retrieved. The following six sound classes are defined: applause, laugh, screaming, music, loud noise, and speech.

Sound classification approach is as follows: At first, GMM for each audio class has been trained on our own collection of sound files (about 2 hours of audio in total). As a front-end, audio signal was parameterized by conventional MFCCs, from which 2-D cepstral matrices are created by applying additional cosine transform along each MFCC within 1sec. block of audio frames. One dimension of 2-D cepstrum is quefrency and the second dimension is modulation frequency, which exposes temporal changes of each MFCC. Audio track of each video in TRECVID 2007 collection is analysed by 2-D cepstrum in 1sec. windows with 0.5 second shift. Then log-likehoods for all 6 GMMs are computed for each audio segment. A segment is assigned to one of the 6 audio classes by applying the following kNN rule on log-likehood vector space created from the labelled training data. If at least half (k/2) of the neighbours belongs to the same class, the segment is assigned to this class, otherwise the segment is not labelled. We applied such kNN based decision rather than maximum log-likehood decision with the aim to obtain the precision as high as possible even if the recall may decrease. Finally, a video shot is assigned as relevant for the certain audio class if at least 2 audio segments, within the shot, are labelled with the given sound class.

4.1.4 Relevance feedback module from QMUL

Relevance feedback (RF) scheme was initially developed for information retrieval systems in which it performs an online learning process aiming at improving effectiveness of search engines. It has been widely applied in image retrieval techniques since the 1990s. RF is able to train the system to adapt its behaviour to users' preferences by involving human into the retrieval process. An image retrieval framework with RF analyse relevant or irrelevant feedback from the user and uses it to predict and learn user's preferences. At the mean time, more relevant image can be successively retrieved.

A system contains RF process is illustrated in Figure 5. It needs to satisfy several conditions:

- Images are presented to the user for his/her feedback, but same images should not be repeated in different iterations.
- The input to the module is relevant/irrelevant information provided by the user on iterative bases.
- The module should automatically learn user's preferences by adapting the system behaviour using the knowledge feedback from the user.

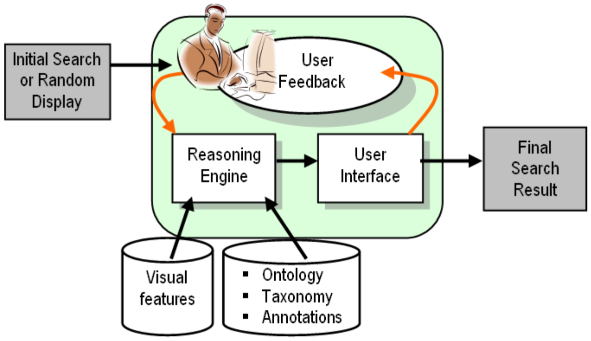

Figure 5: Generalised hybrid content-based image retrieval systems with relevance feedback.

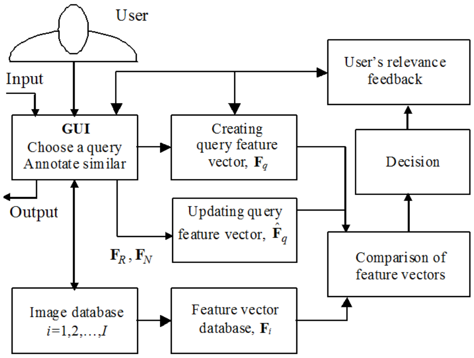
User
Input
GUI
Choose a query
Annotate similar
Output
Creating query feature vector, $\mathbf{F}_q$
Updating query feature vector, $\hat{\mathbf{F}}_q$
$\mathbf{F}_R$, $\mathbf{F}_N$
Image database i=1,2,…,I
Feature vector database, $\mathbf{F}_i$
User's relevance feedback
Decision
Comparison of feature vectors

Figure 6: Block scheme of a CBIR system with RF from U. Belgrade.

A general image retrieval system with RF such as the one displayed in Figure 5 can use any kind of descriptors from low-level information of available content itself to prior knowledge incorporated into ontology or taxonomy.

When a learning approach is considered, many kind of reasoning engine can be used to determine relevant information. There are several common classes of RF modules such as: Descriptive models (e.g. Gaussians, GMMs), Discriminative models (e.g. SVMs, Biased Discriminative Analyses) and Neural networks (e.g. SOMs, Perceptrons).

In our framework, one of the RF modules is implemented by QMUL based on SVM. It combines several MPEG7 or non-MPEG7 descriptors as a cue for learning and classification. SVM is one of the developed supervised learning algorithms. It empirically models a system that predicts accurate responses of unseen dataset based on limited training sets [18].

In submitted runs with QMUL RF, all experiments were conducted using linear SVM for the sake of efficiency. Given the initial search result using visual similarity search or text-based search, users were asked to select at least one positive and one negative examples on screen as feedback. Usually two to five iterations were performed depending on users' preferences, within the time limitation. Four MPEG7 descriptors: Colour Layout, Colour Structure, Edge Histogram and Homogeneous Texture and one non-MPEG7 descriptor: Grey Level Co-occurrence Matrix were used and combined to conduct visual feature based RF [19].

4.1.5 Relevance feedback module from University of Belgrade

In the Laboratory of digital image processing, telemedicine and multimedia (IPTM), Faculty of Electrical Engineering, U. Belgrade, content-based image retrieval (CBIR) module with RF was derived. The module uses low-level image features, such as colour, line directions and texture, for objective description of images. We investigated both global visual descriptors (for a whole image) and local descriptors (for regions) [20], and different combination of visual features mainly based on the MPEG-7 descriptors, including the reduction of feature vector components [21]. In all cases the block scheme of U. Belgrade module was the same, as depicted in Figure 6.

First step in any CBIR system includes the determination of relevant low-level features $j = 1, 2, , J$, describing as best as possible the content of each image i, $i = 1, 2, , I$. Features are expressed by corresponding numerical values, and are grouped into appropriate feature vector $F_i = [F_{i1}, F_{i1}, ...F_{i1}]$ of the length J. Feature vectors were stored in appropriate feature matrix, $F = F_i$, of dimension $I \times J$. Then, the retrieving procedure is based on relatively simple proximity measure (for instance, Euclidean distance, Mahalanobis, or similar) between feature vectors of a query image and images from database.

Components of a feature vector matrix $F = F_i = F(i, j)$, $i = 1, 2, , I$, $j = 1, 2, , J$, are column-wised rescaled with weighted term $W1_j$, according to

$$W1_j = \frac{1}{mean(F_j)} \log_2(std(\frac{F_j}{mean(F_j)}) + 2), j = 1, 2, ..., J. \tag{10}$$

As a similarity measure we used Mahalanobis distance metric, and as a relevance feedback strategy we used a query shifting in combination with the Probabilistic Feature Relevance Learning (PFRL) method [22], and the non-linear modelling capability of the radial basis functions (RBF).

From feature vectors of subjectively annotatad images as relevant R and N, a query feature vector was updated, by using the Rocchio's equation:

$$\hat{F}_q = F_q + \alpha_R(\bar{F}_R - F_q) - \alpha_N(\bar{F}_N - F_q), \tag{11}$$

where F_q is a previous query feature vector, $\hat{F}_q$ is updated vector, and $\bar{F}_R$ and $\bar{F}_N$ are the mean values of feature vectors of R and N images, respectively. Positive constants α_R and α_N determine the influence of R and N images to query vector updating. In our work we associate a one-dimensional Gaussian RBF with each feature vector F_i for images from database

$$S_i(F_i, \hat{F}_q) = \sum_{j=1}^{J} \exp{(-\frac{(F_{ij} - \hat{F}_{qj})^2}{2\sigma_j^2})}, i = 1, 2, ..., I. \tag{12}$$

The magnitude of S_i represents the similarity between the feature vector, F_i, and the modified query $\hat{F}_q$ after user's feedback. Standard deviation, σ_j, determines the slope of Gaussian function and, in particular, reflects to the relevance of j_{th} individual feature.

The functions S_i are then used to determine the image similarity in a new (subjective) search process: the magnitude of functions S_i are stored in descending order, a new set of best matched images is displayed, from which user selects and labels new relevant and irrelevant ones, thus updating RBF and refining the search. The process is repeated until the user is satisfied with retrieved results. Usually, two to three iterations were sufficient.

4.2 Results

We have submitted four runs to the TRECVID 2007 Search task. The four runs in our submission used four different run types respectively. The run types and the results achieved using these runs are illustrated in Table 2 below. It seems that the RF modules were capable of retrieving more relevant shots. However, the achieved scores were not improved due to the limitation of time and the fact that the users did the experiments were relatively inexperienced with the interactive approaches. By analysing our results, it can be observed that our submissions in this year generally outperformed 2006.

Table 2: Evaluation of search task results.

Run type	visual search	visual + text search	visual + text search + RF QMUL	visual + text search + RF U. Belgrade	Mean of 2006
Precision out of total relevant shots	0.075	0.086	0.083	0.069	0.027
Average precision	0.098	0.110	0.096	0.078	0.023

5 Rushes Task

The rushes task submission is based on a video summarisation and browsing system comprising two different interest curve algorithms and three features. This system is a result of joint work of TU Delft, QMUL, LaBRI and VICOMTech.

5.1 Interesting moment detector by TU Delft

We approach the modelling of the *experience* of a rushes video by extending our previous work on arousal modelling [23]. Based on a number of audio-visual and editing features, the effect of which on a human viewer can be related to how that viewer *experiences* different parts of the audiovisual material, we model the arousal time curve that represents the variations in experience from one time stamp to another. High arousal values ideally represent the parts of the video with high excitement, as compared to more-or-less *serene* parts represented by low arousal values. The obtained curve can be used to automatically extract the parts of the unedited video that are best capable of eliciting a particular *experience* in the given total duration. We expect these high arousal parts to be more significant in the video and they should be shown to the user in the first place.

5.2 Interesting moment detector by QMUL

The frames are firstly clustered using the Normalised Cuts algorithm, *Ncut*. This algorithm was first introduced by Shi and Malik in [24] as a heuristic algorithm aiming to minimise the *Normalised Cut* criterion between two sets, defined as:

$$NCut(A,B) = Cut(A,B)\,(1/VolA + 1/VolB) \tag{13}$$

where $cut(A,B) = \sum_{i \in A, j \in B} w_i, j$, and w_i, j are pairwise similarities between points i and j:

$$w_{i,j} = e^{-\frac{d(i,j)^2}{2\sigma^2}} \tag{14}$$

where $d(i,j)$ is a distance over the set of low-level feature vectors. Originally this approach was created to solve the problem of perceptual grouping in the image data. The first step in our algorithm is to create the similarity matrix. Instead of analysing the video on a key frame level, we use a predefined ratio of frames in order to stimulate the block structure of the similarity matrix. The main task in the video summarisation is to properly cluster scenes, and then to analyse clusters in order to find most informative representatives. Spectral algorithms use information contained in the eigenvectors of data affinity matrix to detect structures. Given a set of data points, the similarity matrix is defined as matrix W with elements $w_{i,j}$. Let D be a $N \times N$ matrix with values $d_i = \sum_{j \in I} w_i j$, $i \in [1,N]$ on its diagonal. Then Laplacian matrix of the given dataset is defined as:

$$L = D - W \tag{15}$$

After creating the similarity matrix and solving the generalised eigensystem:

$$Lx = \lambda Dx \tag{16}$$

with λ being an eigenvalue and x being corresponding eigenvector, the next step is to determine the number of clusters in the video, k. Automatic determination of the number of clusters is not a trivial task. Every similarity matrix have a set of appropriate number of clusters depending on the choice of the parameter σ. For automatic detection of number of clusters for fixed σ, we use the results of matrix perturbation theory.The matrix perturbation theory states that the number of clusters in a dataset is highly dependent on the stability of eigenvalues/eigenvectors determined by the eigengap , defined as:

$$\delta_i = |\lambda_i - \lambda_{i+1}| \tag{17}$$

with λ_i and λ_{i+1}, being two consecutive eigenvalues of (16). The number of clusters k is then found by searching for the maximal eigengap over a set of eigenvalues:

$$k = \left\{ i | \delta_i = \underset{j=1\cdots N}{max}(\lambda_j) \right\} \tag{18}$$

After the number of clusters k is found, $N \times k$ matrix X is created by stacking the top k eigenvectors in columns. Each row of X corresponds to a point in the dataset and is represented in a k-dimensional Euclidian space. Finally, k clusters are obtained by applying the K-means algorithm over the rows of X. Results of the k-means algorithm are clusters that give importance information for various applications. Scenes that contain different events result in non continuous clusters detected by the k-means algorithm. These non constant clusters correspond to the scenes in the video. In order to properly detect these scenes, frames belonging to the same clusters, separated by the frames of other clusters, should be merged in one scene together with frames that lay between them on the time axis. This is done by analysing the structure of the clusters obtained by the k-mean algorithm. Let $I(i)$ be the cluster indicator of the frame i, with

$i \in [1, N]$. First frame of the cluster I is i_1, and i_b is the first frame of the cluster I with $I(i_b) \neq I(i_b + 1)$ and all frames between i_1 and i_b belonging to the same cluster. Finally clusters are merged by putting all frames between i_b and i_e to the same cluster, where i_e is defined as:

$$i_e = \max_{k=1 \cdots t_{tr}} \{k | I(i_b) = I(i_b + k)\} \tag{19}$$

where t_{tr} is experimentally determined threshold. Now each cluster is supposed to be corresponding to one scene in the video. These scenes are further used as basic units for important event detection in the following steps.

5.3 Features by LaBRI and VICOMTech

In rushes videos, there are some parts that are meaningless in the final edited movie and thus should not appear in the summary. We can distinguish two kinds of such parts: unwanted frames which are generally frames with nothing visible and unscripted parts showing the movie crew setting up a scene for example.

5.3.1 Unwanted frames

Unwanted frames are composed in particular of frames with uniform colour (often all black or gray), or with colour bars (see Figure 7). According to our observations, such frames appear randomly during the rushes, and we may still hear sound from the scene in the background. In order to detect these unwanted frames, we compute a colour histogram on each channel of a frame, in RGB format. We then sum the peaks of these histograms, and classify the frame as unwanted if this sum is superior to a given threshold. We then use a median filter (of width 5) to filter this result, as wrong frames most often last several seconds.

For performance reason, we apply this detection only at I-frame resolution and interpolate the filtered results for P-frames.

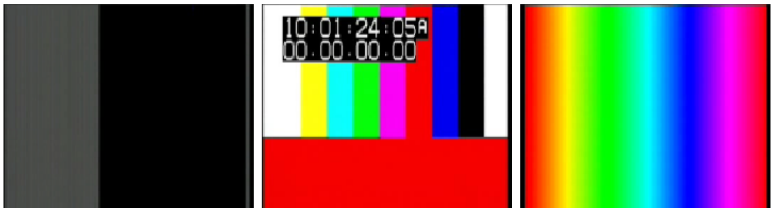

Figure 7: Unwanted frames: a) grey/black frame; b) sharp color bars; b) diffuse color bars.

This algorithm seems to work pretty well on totally black or gray frames and on sharp colour bars (Figure 7 b)on the *devel* movies. But it is not appropriate for diffuse colour bars found on some videos. Moreover, this method may also falsely detect some scripted scenes, very dark scenes in particular. But we believe that scenes with so few colours are most often not very understandable and thus does not need to be in the summary.

5.3.2 Human detection

On of the extracted features, is human detection on frames. Indeed, we believe a human is one of the more recognisable object in a video, especially for a human summariser.

We use detection of skin colour to detect human presence in frames. Our human detection algorithm has several steps. First, we use OpenCV [25] to detect front-facing faces, only on I frames for performance reason. We apply a simple geometric filter to remove too small or too big faces, then a we apply a temporal median filter on bounding boxes of detections. In a second pass, we use the algorithm described in [26]. We first train a colour detector on previous detected faces to extract a colour model for faces, specific to this movie. then we process a second time I frames of the whole movie to detect areas corresponding to the colour model. We apply again our temporal median filter on bounding boxes of detections. finally, we interpolate our results to P-frame temporal resolution. The computed colour model is highly dependent of the precision of the first step. It seems that we could improve our results with a better parametrisation of OpenCV.

5.3.3 Camera motion

Another extracted feature is camera motion. Indeed, camera motion is often used by the director to highlight a significant event in a movie. So we believe that a part where a camera motion occurs is rather important. Moreover, “camera event” is reported in the summary ground truth instructions as one of the event to note for the summariser.

We use the algorithm described in [27]. First we estimate the global camera motion, extracting only motion vectors from P-frames of MPEG compressed stream. Then we use a likelihood significance test of the camera parameters to classify specific camera motions (pan, zoom, tilt).

On the provided rushes videos, it seems that many camera motions occur during unscripted parts of scenes, during the scene set up in particular. However, this information may still be discriminatory when other features are unavailable.

5.4 Merging and Layout

The merging of the features is performed in a heuristic manner. Once the frames in the system are clustered, the importance of each cluster and the points in each cluster where the importance is highest are determined. The summary of the video is prepared by merging the parts from each cluster around the maximum importance point. The length of the cluster summaries are directly proportional to the cluster importance values. The frame importance is a weighted sum of excitement level, detected number of faces and camera motion type whose weights are set after user tests. The cluster importance is the average of frame importance values in corresponding cluster.

5.5 Results

Thanks to the success of the clustering algorithm, our system performed as the second best algorithms in terms of minimising the duplications. Also it is scored above average as easy to understand. On the other hand, our system has performed below average in terms of fraction of inclusions. Since our system is not designed to detect and separate events but relies on some low level properties of video to detect the important sections, it may miss the parts with some defined events but without high excitement, camera motion or faces.

References

[1] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR ’06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[2] J. Calic and all. Cost292 experimental framework for trecvid 2006. November 2006.

[3] Y. Yusoff, W.J. Christmas, and J.V. Kittler. Video shot cut detection using adaptive thresholding. In *BMVC00*, 2000.

[4] N. Adami and R. Leonardi. Identification of editing effect in image sequences by statistical modeling. In *Picture Coding Symposium*, pages 0–4, Portland, Oregon, U.S.A., April 1999.

[5] S.U. Naci and A. Hanjalic. Low level analysis of video using spatiotemporal pixel blocks. In *Lecture Notes in Computer Science*, volume 4105, pages 777–784. Springer Berlin / Heidelberg, 2006.

[6] C. Petersohn. Dissolve shot boundary determination. In *Proc. IEE European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, pages 87–94, London, UK, 2004.

[7] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on PAMI*, 8:888–905, 2000.

[8] E. Spyrou and Y. Avrithis. A region thesaurus approach for high-level concept detection in the natural disaster domain. In *2nd International Conference on Semantics And digital Media Technologies (SAMT)*, 2007.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 886–893, June 2005.

[10] D. Gokalp and S. Aksoy. Scene classification using bag-of-regions representations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Beyond Patches Workshop*, Minneapolis, Minnesota, June 23, 2007.

[11] I. Kompatsiaris and M. G. Strintzis. Spatiotemporal segmentation and tracking of objects for visualization of videoconference image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(8):1388–1402, December 2000.

[12] Krishna Chandramouli and Ebroul Izquierdo. Image classification using self organising feature map and particle swarm optimisation. In *Proceedings of 3rd International Conference on Visual Information Engineering*, pages 313–316, 2006.

[13] T. Kohonen. The self organizing map. *Proceedings of IEEE*, 78(4):1464–1480, September 1990.

[14] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.

[15] V. Mezaris, H. Doulaverakis, S. Herrmann, B. Lehane, N. O'Connor, I. Kompatsiaris, and M. G. Strintzis. Combining textual and visual information processing for interactive video retrieval. In *in proceedings of TRECVID 2004*, Gaithersburg, MD, USA, 2004.

[16] A. Gutmann. R-trees: a dynamic index structure for spatial searching. In *proceedings of ACM International Conference on Management and Data (SIGMOD'88)*, Siena, Italy, 1988.

[17] S.E. Intille and K. Sparck Jones. Simple, proven approaches to text retrieval. Technical Report UCAM-CL-TR-356, 1997.

[18] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[19] D. Djordjevic and E. Izquierdo. Kernel in structured multi-feature spaces for image retrieval. *Electronics Letters*, 42(15):856–857, 2006.

[20] S. Rudinac, M. Ućumlić, M. Rudinac, G. Zajić, and B. Reljin. Global image search vs. regional search in CBIR systems. In *proceedings of Conf. WIAMIS 2007*, Santorini, Greece, 2007.

[21] G. Zajić, N. Kojić, V. Radosavljević, M. Rudinac, S. Rudinac, N. Reljin, I. Reljin, and B. Reljin. Accelerating of image retrieval in CBIR system with relevance feedback. *Journal of Advances in Signal Processing*, 2007.

[22] J. Peng, B. Bhanu, and S. Qing. Probabilistic feature relevance learning for content-based image retrieval. *Computer Vision and Image Understanding*, (1/2), 1999.

[23] L.-Q. Xu A. Hanjalic. Affective video content representation and modeling. *IEEE Transactions on Multimedia*, February 2005.

[24] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8), 2000.

[25] Opencv. http://opencvlibrary.sourceforge.net, 2007.

[26] A. Don, L. Carminati, and J. Benois-Pineau. Detection of visual dialog scenes in video content based on structural and semantic features. In *International Workshop on Content-based Multimedia Indexing (CBMI) 2005*, Létonie (Tampere), 2005.

[27] P. Kraemer, J. Benois-Pineau, and M. Gràcia Pla. Indexing camera motion integrating knowledge of quality of the encoded video. In *Proc. 1st International Conference on Semantic and Digital Media Technologies (SAMT)*, December 2006.