COST292 experiments for TRECVID 2006

J. Ćalić* N. Campbell* P. Krämer† J. Benois-Pineau† S. Vrochidis‡ C. Doulaverakis‡
V. Mezaris‡ I. Kompatsiaris‡ E. Spyrou§ G. Koumoulos§ Y. Avrithis§ S. Aksoy¶
A. Dalkilic‖ A. Saracoglu** A. Alatan** Q. Zhang†† E. Izquierdo†† U. Naci‡‡ A. Hanjalic‡‡

Abstract

In this paper we give an overview of the four TRECVID tasks submitted by COST292, European network of institutions in the area of semantic multimodal analysis and retrieval of digital video media. Initially, we present shot boundary evaluation method based on results merged using a confidence measure. The two SB detectors user here are presented, one of the Technical University of Delft and one of the LaBRI, University of Bordeaux 1, followed by the description of the merging algorithm. The high-level feature extraction task comprises three separate systems. The first system, developed by the National Technical University of Athens (NTUA) utilises a set of MPEG-7 low-level descriptors and Latent Semantic Analysis to detect the features. The second system, developed by Bilkent University, uses a Bayesian classifier trained with a "bag of subregions" for each keyframe. The third system by the Middle East Technical University (METU) exploits textual information in the video using character recognition methodology. The system submitted to the search task is an interactive retrieval application developed by Queen Mary, University of London, University of Zilina and ITI from Thessaloniki, combining basic retrieval functionalities in various modalities (i.e. visual, audio, textual) with a user interface supporting the submission of queries using any combination of the available retrieval tools and the accumulation of relevant retrieval results over all queries submitted by a single user during a specified time interval. Finally, the rushes task submission comprises a video summarisation and browsing system specifically designed to intuitively and efficiently presents rushes material in video production environment. This system is a result of joint work of University of Bristol, Technical University of Delft and LaBRI, University of Bordeaux 1.

1 Introduction

This paper describes collaborative work of a number of European institutions in the area of video retrieval joined under a research network COST292. COST is an intergovernmental network which is scientifically

*J. Ćalić and N. Campbell are with Dept. of Computer Science, 2.11 MVB, Woodland Road, University of Bristol, Bristol BS8 1UB, UK, {janko,campbell}@cs.bris.ac.uk

†P. Krämer and J. Benois-Pineau are with LaBRI, University of Bordeaux 1, 351 cours de la Libération, F-33405 Talence, {petra.kraemer,jenny.benois}@labri.fr

‡S. Vrochidis, C. Doulaverakis, V. Mezaris 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,doulaver,bmezaris,ikom}@iti.gr

§E. Spyrou, G. Koumoulos and Y. Avrithis are with Image Video and Multimedia Laboratory, National Technical University of Athens, 9 Iroon Polytechniou Str., 157 80, Athens, Greece

¶S. Aksoy is with Department of Computer Engineering, Bilkent University, Bilkent, 06800, Ankara, Turkey

‖A. Dalkilic is with Department of Computer Engineering, Hacettepe University, Cankaya, 06532, Ankara, Turkey

**A. Saracoglu, A. Alatan are with Department Of Electrical and Electronics Engineering, Middle East Technical University, 06531, Ankara, Turkey

††Q. Zhang and E. Izquierdo are with Department of Electronic Engineering, Queen Mary, University of London, Mile End, London E1 4NS, UK, {qianni.zhang, ebroul.izquierdo}@elec.qmul.ac.uk

‡‡U. Naci, A. Hanjalic are with Delft University of Technology, Mekelweg 4, 2628CD, Delft, The Netherlands

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 multimodal 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. Therefore, this year our group has submitted results to all four tasks. The following sections bring details of applied algorithms and their evaluation.

2 Shot Boundary Detection Task

With the objective to optimally utilise results of several shot boundary (SB) detection tools developed by the COST292 participants, we merged the results of two SB detectors using a confidence measure. Thus, we will first introduce the two SB detectors, that one of the Technical University of Delft and that one of the LaBRI, University of Bordeaux 1, and secondly the method to merge the results of the two SB detectors in the decision space.

2.1 SB Detector by the TU Delft

The proposed method introduces the concept of *spatiotemporal block based analysis* for the extraction of low level events. The proposed 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 detailed description of the system can be found in [1].

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.

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

Here, I is the pixel intensity function and $\vec{\mathbf{v}} = (v_x, v_y)$, i.e. 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):

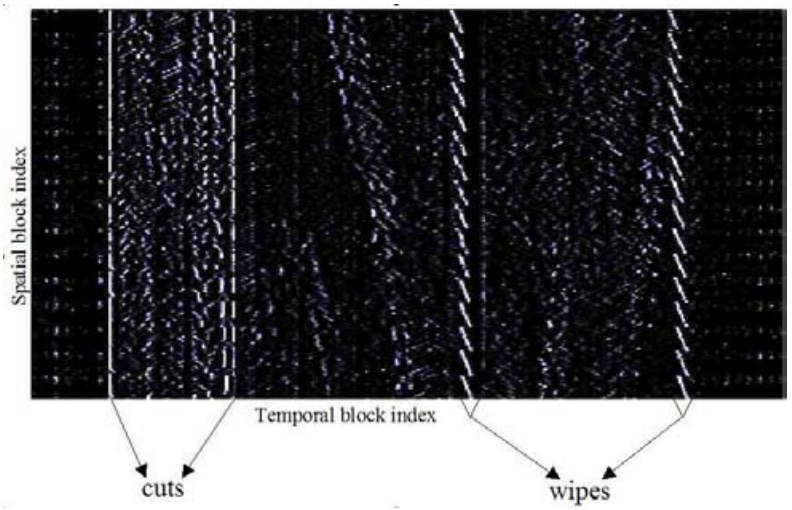

Figure 1: An illustration of confidence values for block based abrupt changes.

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

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

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 $F_1(i,j,K) \cdot F_1(i,j,K)$ values over all spatial indices (i,j) at the corresponding time 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 1 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.

2.2 SB Detector by the LaBRI

The SB detector developed by the LaBRI utilises the "Rough Indexing Paradigm" i.e. we work on compressed video only in I/P resolution. The SB detector we used for TRECVID 2006 is an improved version of the algorithm presented at TRECVID 2004 and 2005 [2].

The SB detector works separately on I-frames and P-frames. The detection on P-frames is based on the temporal difference of intra-coded macroblocks ΔQ and the variation of global motion parameters V. Therefore, the affine 6 parameter model is estimated from the encoded motion compensation vectors.

The mix function of the detector we previously developed [2], combining $\Delta Q(t)$ and $V(t)$ into one value, adsorbs the local maximum if one of these values is very small.

Therefore, we defined a new mix function $M(t)$ normalised in $[0, 1]$:

$$M(t) = \text{sign}(\tilde{\Delta}Q(t)) \cdot (1 - (|1 - \tilde{\Delta}Q(t)| \cdot \tilde{V}(t))) \quad (8)$$

Here, $\tilde{\Delta}Q$ and $\tilde{V}$ are respectively the normalized values of ΔQ and V. Since the translational parameters of the global motion model vary very much and thus cause a lot of overdetections, we take account only of the affine parameters in the computation of V. Then, the SB detection is based on a local maximum search on $|M(t)|$ and a change of the sign of $M(t)$. A detected SB is classified as gradual transition, if other local maxima occur in the neighboring P-frames, otherwise it is classified as a cut. The confidence measure for P-frames is based on the error probability on a Gaussian distribution of the measure (8).

The detection method for I-frames reuses the global motion models of the SB detection on P-frames. It is used to calculate the histogram intersection of the DC image of the current I-frame and the motion compensated DC image of the previous I-frame. In order to detect a SB, the values of the histogram intersection are thresholded. Then, a detected SB is classified as gradual transition if one of the neighboring I-frames has a strong histogram intersection value too. Otherwise it is classified as a cut. The confidence measure for a detection on an I-frame is proportional to the margin between the histogram intersection value and the detection threshold.

2.3 Merging

The merging was performed under the basic assumption that the SB detector of the TU Delft achieves a higher precision and recall, since the SB detector of the LaBRI works in the compressed domain only in I/P resolution. For each detector, the SB detection results are characterized by a confidence measure. In the merging process, we use both confidence measures and privilege the SB detector of the TU Delft.

Let $B_D = \{b_D\}$ the set of SB detections of the TU Delft, $B_L = \{b_L\}$ the set of SB detections of the LaBRI, c_D and c_L the associated confidence measures, and C_D and C_L two thresholds with $C_D < C_L$. If a SB $b_D \in B_D$ does not intersect any SB $b_L \in B_L$, and if $c_D > C_D$, then b_D is retained as a detection result. If a SB $b_L \in B_L$ does not intersect any SB $b_D \in B_D$, and if $c_L > C_L$, then b_L is retained as a detection result. In the case of $b_D \cap b_L$, b_D is retained as a detection result.

2.4 SB Detection Results

Finally, 10 runs have been submitted by COST292. They are composed as follows: Four merged runs have been submitted as COST292-1 to COST292-4 with respectively a recall and precision of 65.14/46.69, 65.14/46.69, 65.19/46.70, and 64.29/64.00. Four individual runs of the TU Delft have been submitted as COST292-5 to COST292-8 with a recall and precision of 65.16/45.07, 64.29/62.76, 66.49/77.21, and 67.17/74.48. Two individual runs of the LaBRI have been submitted as COST292-9 and COST292-10 with a recall and precision of 61.80/51.88 and 55.64/56.81.

3 High-level feature extraction

COST292 participated to the high-level feature extraction task with three separate systems. The first system, developed by the National Technical University of Athens (NTUA) is described in Section 3.1. The second system, developed by Bilkent University is described in Section 3.2. Finally, the third system by the Middle East Technical University (METU) is described in Section 3.3.

3.1 Feature extractor from NTUA

In this section we present our approach for the detection of certain high-level concepts in the TRECVID video sequences corresponding to the run COST292R1. We selected and tried to detect the following 7

features: *desert*, *vegetation*, *mountain*, *road*, *sky*, *fire-explosion* and *snow*. Our approach used the provided extracted keyframes of each video sequence.

The first step of our method was to select an appropriate low-level description of each keyframe. A description based on the MPEG-7 standard was selected, that combined both color and texture features of each keyframe. A K-means clustering method is applied on the RGB values of the keyframe, dividing it in K regions. The centroid of each region is its dominant color. We also extract the MPEG-7 Homogeneous Texture Descriptor (HTD) [3] of each region, in order to capture its texture properties efficiently. Then we scale and merge the aforementioned visual descriptions of the keyframe into a single vector.

In the next step of our method we create the "region thesaurus" containing the "region types". This thesaurus is actually a dictionary and each region type is a word of the dictionary. It contains the visual descriptions of certain image regions which will be used as prototypes. There have been examined two methods for the creation of this thesaurus. The first one uses the *subtractive clustering* [4] method. This way, both the number of the clusters and their corresponding centroids are estimated and each centroid is a word of the dictionary. The second method uses a predefined number of words. After some experiments this number was set to 100 as it led to fast yet effective performance.

We should clarify here that each region type may or may not represent a high-level feature and each high-level feature may be represented by one or more region types. For example, the concept *desert* can have more than one region types differing in i.e. the color of the sand, each represented by a region type of the thesaurus. Moreover, in a cluster that may contain synonyms from the semantic entity i.e. *sky*, they can be mixed up with parts from i.e. *sea*, if present in an image because of the obvious similarity in their low-level visual features.

For each keyframe, we form a model vector with dimensionality equal to the number of concepts that constitute the thesaurus. Having calculated the distance of each region (cluster) of the image to all the words of the constructed thesaurus, the model vector that semantically describes the visual content of the image is formed by keeping the smaller distance for each high-level concept. More specifically, let: $d_i^1, d_i^2, ..., d_i^j, i = 1, 2, 3, 4$ and $j = N_C$, where N_C denotes the number of words of the lexicon and d_i^j is the distance of the i-th region of the clustered image to the j-th region type. Then, the model vector D_m is formed in the way depicted in equation 9.

$$D_m = [min\{d_i^1\}, min\{d_i^2\}, ..., min\{d_i^{N_C}\}], i = 1, 2, 3, 4 \tag{9}$$

Then we follow a Latent Semantic Analysis [5] approach as in [6]. We construct the co-occurrence matrix of region types in given keyframes of the training set in contexts (region types in the thesaurus). The distance function we use to compare a given region type with one of the thesaurus, in order to assign each region of the image to the correct prototype region is a linear combination of a Euclidean distance for the dominant color and the MPEG-7 standardized distance for the HTD.

After of 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. The input of the NN is the model vector in the semantic space and the output represents the distance of each region to the corresponding semantic concept.

3.2 Feature extractor from Bilkent

The system developed by Bilkent University uses a Bayesian classifier trained with a "bag of subregions" for each keyframe. This approach first divides each keyframe into subregions using a fixed grid. Then, the resulting subregions are assigned a cluster label based on low-level features. Each keyframe is represented as a list of these labels. We use two separate models to learn the contributions of these subregions to different classes in a Bayesian classifier. The details of these steps are described below.

3.2.1 Image representation

We model spatial content of images using grids. The low-level features based on color, texture and edge are computed individually on each grid cell of a non-overlapping partitioning of 352×240 frames into 5 rows and 7 columns. The color features include histograms of HSV values, texture features include means and standard deviations of Gabor responses, and edge features include histograms of Canny-based edge orientations. After feature extraction, the ISODATA algorithm is used to cluster all feature vectors for all subregions, and a cluster label is assigned to all of the subregions in all keyframes. In the experiments using the TRECVID 2005 data, the final number of clusters was found as 115 by the ISODATA algorithm. Finally, each keyframe is associated with a list of cluster labels corresponding to a "bag of subregions" representation.

3.2.2 Bayesian classifier

Given the list of labels $\{x_1, \ldots, x_m\}$ for a keyframe with m subregions, the goal is to classify this keyframe using the posterior probability $p(w_j|x_1, \ldots, x_m)$ where $w_j, j = 1, \ldots, c$ represents the classes. Assuming equal priors for all classes, the classification problem reduces to the computation of class-conditional probabilities $p(x_1, \ldots, x_m|w_j)$. To simplify this class-conditional probability that would normally have k^m possible terms when estimated jointly, we assume that each subregion is independent of others given the class and use $p(x_1, \ldots, x_m|w_j) = \prod_{i=1}^{m} p(x_i|w_j)$.

We use the bag of subregions representation in two settings for classification. In the first setting, the labels in the representation are assumed to be independent of the corresponding subregion locations. In other words, the probability of subregion x_i having label u is computed as $p(x_i = u|w_j) = p_{ju}$ where $j = 1, \ldots, c$ and $u \in \{1, \ldots, k\}$. Note that p_{ju} is independent of i. We model the class-conditional densities using multinomial distributions. Then, the maximum likelihood estimate of p_{ju} becomes

$$\hat{p}_{ju} = \frac{n_{ju}}{n_j} \tag{10}$$

where n_{ju} is the number of subregions with label u in the example images for class j, and n_j is the total number of subregions in the examples for j. In this model, a total of k parameters need to be estimated for each class.

In the second setting, the model is sensitive to the subregion location. Therefore, the probability of subregion x_i having label u is computed as $p(x_i = u|w_j) = p_{jiu}$ where $j = 1, \ldots, c$, $i = 1, \ldots, m$ and $u \in \{1, \ldots, k\}$. Then, the maximum likelihood estimate of p_{jiu} becomes

$$\hat{p}_{jiu} = \frac{n_{jiu}}{n_{ji}} \tag{11}$$

where n_{jiu} is the number of subregions at location i with label u in the example images for class j, and n_{ji} is the total number of subregions at location i in the examples for j. In this model, a total of mk parameters need to be estimated for each class.

We have trained these models using the TRECVID 2005 data and common annotation for six classes: snow, vegetation, waterscape, sky, mountain and outdoor. After evaluation of the classifiers using these data, the second model that is sensitive to the subregion location was applied to the TRECVID 2006 test data and was submitted as the run COST292R2.

3.3 Feature extractor from METU

For the indexing and management of large scale news video databases, an important tool is textual information within the digital media. Such information, for example, can be used to index any video database quite efficiently and effectively. Speaker information, location, date/time, score results and etc. can be queried more thoroughly. In our first participation to TRECVID we utilized this concept in order to extract high-level features. In our work we only aimed to extract two high-level semantic features, namely

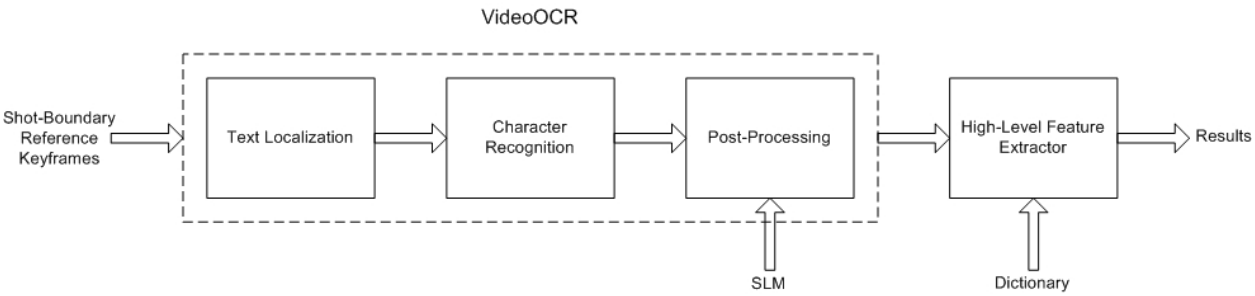

Figure 2: Block diagram of the METU system.

"Government-Leader" and "Corporate-Leader" the choice is mainly because of the reason that extracting these features from the textual information that is present in news video would be more effective and effortless compared to some complicated methods, since the presence of a leader in a shot is stressed by the overlay text containing an informative information from the textual taxonomy of the feature.

Our system (Figure 2) mainly consists of two parts; first part works as a VideoOCR which extracts textual information from keyframes, and the second part extracts high-level semantic features from these extracted textual information. In the VideoOCR section first a Text Localization is employed in which a feature extraction and a classifier are utilized to extract minimum bounding rectangles. After the localization step a neural network based Character Recognition part determines textual information present in each of the extracted bounding rectangles. At the last step of the VideoOCR segment a Statistical Language Model of English is utilized to rectify probable errors at the character recognition step. For the extraction of high-level features although textual taxonomy was aimed to be used, instead a simple dictionary of a feature is utilized with the conjunction of Levenshtein Distance method for the decision process. This dictionary, in addition to descriptive words of features such as "president", "prime-minister", "chancellor of the exchequer" and etc., is constructed from some of the names of the leaders such as "George W. Bush", "Tony Blair" and others for "Government-Leader" feature. Lastly, neither the training of the classifier nor the extraction of the language model is conducted on the TRECVID data, and the system used shot boundary reference keyframes as input.

In the evaluation phase of the High-Level Feature extraction task only "Corporate-Leader" has been included, and as a result our contribution in run COST292R3 has only been that feature. In our further contributions to TRECVID we plan to use textual information approach combining with other basic feature extraction methods to extract the high-level semantic features and also we plan to increase our system's genericness.

4 Interactive Search

In this section, a description of the search platform integrated at ITI for our participation in the TRECVID 2006 Search task is presented. The developed system is an interactive retrieval application, depicted in Figure 3, combining basic retrieval functionalities in various modalities (i.e. visual, audio, textual) with a user interface supporting the submission of queries using any combination of the available retrieval tools and the accumulation of relevant retrieval 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
- Relevance feedback module

Figure 3: User interface of the interactive search platform

The search application combining the aforementioned modules is built on web technologies, specifically php, JavaScript and a mySQL database, providing a GUI for performing retrieval experiments over the internet (Figure 3). Using this GUI, the user is allowed to employ any combination of either all the supported retrieval functionalities or a subset of them to submit a query, view the retrieval results (keyframes) ordered according to rank, and eventually store the identity of those considered to be relevant results for the given query. The latter is made possible using a storage structure that mimics the functionality of the shopping cart found in electronic commerce sites. This way, the user can repeat 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. The latter is set to 15 minutes for our experiments, in accordance with TRECVID guidelines. A detailed description of each retrieval module of this application is given in the following section.

4.1 Retrieval Module Description

4.1.1 Visual similarity search

In the developed application, visual similarity search is realized using MPEG-7 XM and its extensions. The MPEG-7 XM supports two main functionalities, i.e. (a) extraction of a standardized Descriptor for a collection of images and (b) retrieval of images of a collection that are similar to a given example, using the previously extracted standardized Descriptor and a corresponding matching function. Employed extensions to the MPEG-7 XM include the MultiImage module, effectively combining more than one MPEG-7 descriptor, the XM Server, which forces the original command-line MPEG-7 XM software to constantly run as a process in the background so as not to repeat decoding of binary descriptor files during each query, and the Indexing module, which employs an indexing structure to speed up query execution. Visual similarity search using MPEG-7 XM and its extensions is presented in more detail in [7].

4.1.2 Textual information processing module

Text query is based on the video shot audio information. The text algorithm integrated in the search platform is the BM25 algorithm, which incorporates both normalized 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 [8] to produce good results.

4.1.3 Audio filtering

Audio-based filtering was implemented by exploiting audio features extracted for each shot, indicating the presence of noise, speech and music in the shot. Using these three relatively high-level audio features, the user is allowed to specify whether the results of his query should have any specific audio characteristics, e.g. include only shots where speech is present, not include shots with music, etc. This filtering is primarily used in combination of visual similarity or text-based retrieval.

The audio information filtering works in compressed domain on audio portion of MPEG-1 bitstream. The procedure of audio signal processing was as follows.

Each video file from the TRECVID collection was demultiplexed. Then, only the scalefactors of the subbands, which fall into the frequency range 0-5.5 kHz, were extracted from MPEG-1 audio layer II bitstreams (mp2). (Note, since the bandwidth B of each mp2 subband depends on the sampling frequency, B=0.5 fs/32, the number of the scalefactors/subbands extracted varies. The first 8 subbands were extracted if fs=44.1 or 48 kHz, but the first 11 subbands if fs=32 kHz.

The stream was split into temporal segments (clips) of 1.3 second length. Each clip was described by K-by-L matrix of the scalefactors. K and L correspond to the number of subbands and mp2 frames respectively. Again, the number of mp2 frames in one clip varies since the frame resolution (or scalefactor-level resolution) depends on the sampling frequency, res=32x12/fs. The sizes of the matrices are 8-by-162, 8-by-150, or 11-by-108 if fs=48, 44.1, or 32 kHz respectively. For silence detection, an energy level of the signal was determined by superposition of all relevant scalefactors. The clips, in which the level was below the threshold, were assigned as silent.

On each 3 subsequent clips (i.e. 3.9 second analysis window), the following two temporal features were extracted: MaximumPeakDuration - duration of the widest peak within the analysis window, PeakRate - number of energy peaks per second. Following approach introduced in [Jar01], energy peaks were extracted by simple thresholding of the sum of relevant scalefactors. The scalefactors of the lowest subband were excluded from the energy contour computation.

To detect the occurrence of rhythmic pulses, a long-term autocorrelation function was applied to a temporal sequence of the scalefactors in each subband. This analysis is applied on a windows formed from 5 subsequent clips (i.e. 6.5 second analysis window). Should a peak occur in the function, the magnitude of this peak would reflect the level of rhythm in the signal. The maximum value of these peaks over all K normalized autocorrelation functions within the analysis window was chosen as the third low-level audio feature called RhythmMetric [Jar02].

By sliding the analysis windows clip-by-clip, each audio stream was described by 3 low-level features with 1.3 second resolution. The clips were divided into 4 classes, namely silence, speech, music, noise, by rule-based classifier as follows.

```
if ClipEnergy<TH1 then clip=silence
elseif MaximumPeakDuration<TH2 AND PeakRate>TH3 AND PeakRate<TH4 AND RhythmMetric<TH5
then clip=speech
elseif RhythmMetric>TH6 then clip=music else clip=noise
```

The thresholds TH were determined by trial and error examination of various parts of video recordings from the TRECVID2006 collection and own sources. Here in the filter, the following values were applied: TH1 = 0.25 of average energy over the recording; TH2 = 1.5 sec.; TH3 = 1.5 per sec.; TH4 = 5.9 per sec.; TH5 = 0.56; TH6 = 0.4.

Then relevances for each audio class were determined on video-frame level by temporal mapping of the clips into relevant video frames. For each video shot, speech/music/noise relevances were assigned. The relevances were set to one (true) if at least 39, 57, or 79 video frames were assigned as relevant within short shots (up to 120 frames), middle-length shots (up to 180 frames), and long shots (longer than 180 frames) respectively. Otherwise they were set to zero.

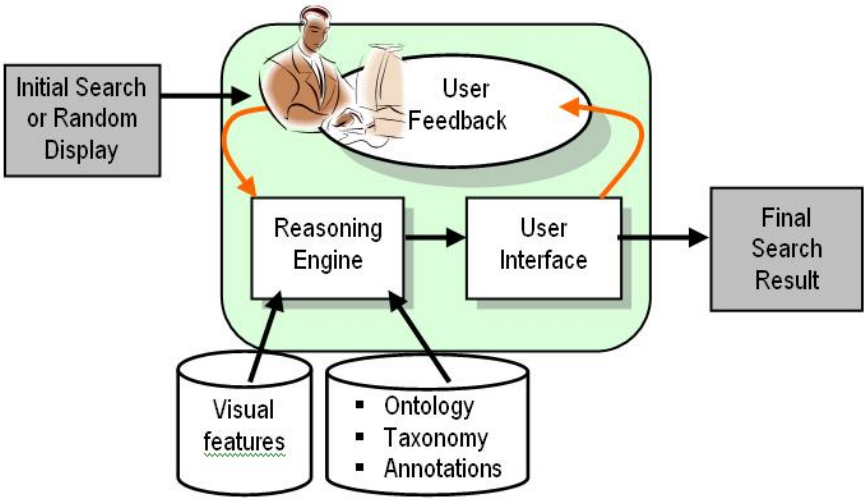

Figure 4: Generalized hybrid content-based image retrieval systems with relevance feedback

4.1.4 Relevance feedback module

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. Relevance feedback 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 relevance feedback collects relevant or irrelevant information provided by the user and uses it to predict and learn user's preferences [9]. At the mean time, more relevant image can be successively retrieved.

A system contains RF process is illustrated in Figure 4. 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 and/or 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.

A general image retrieval system with RF such as the one displayed in Figure 4 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. Some common classes of RF modules are:

- Descriptive models (e.g. Gaussians, Gaussian Mixture Models (GMM)).
- Discriminative models (e.g. Support Vector Machines (SVM), Biased Discriminative Analyses (BDA)).
- Neural networks (e.g. Self Organizing Maps (SOM), Perceptrons).

4.1.5 Support Vector Machines for Relevance Feedback

In our framework a RF module based on SVM is implemented which 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 [10].

Given a set of training data generated by an unknown probability distribution $P(x, c)$, x is a N-dimensional data sample and c is an appropriate label defining the membership of a data sample to a particular class.

$$(x_1, c_1), (x_2, c_2), ..., (x_m, c_m) \in R^N \times \{-1, +1\}$$

Being general, when m training examples are provided, there can be at most m classes. But in binary classification scenarios there are only two classes, which is the simplest case. The aim is to find a function $f : R^N \rightarrow \{-1, +1\}$ that would correctly classify the unseen testing examples generated out of the same probability distribution as the training set. In vector space the separating function is a hyperplane that can separate the vector data, which takes the form

$$(w \cdot x) - b = 0, w \in R^N, b \in R \tag{12}$$

In (12) the vector w points perpendicular to the separating hyperplane. A margin exists on each side of the hyperplane between the hyperplane and the closest x. b is the offset parameter controlling the width of the margins. The corresponding decision function classifier is denoted as:

$$f(x) = sgn((w \cdot x_i) - b) \geq 1, i = 1, ..., m$$

The separating hyperplane is optimal if it separates a set of patterns without error and maximizes the margins. The optimal solution can be obtained from a following optimization problem:

$$\min \tfrac{1}{2}\|w\|^2, under\ the\ constraint : c_i((w \cdot x_i) - b) \geq 1, i = 1, ..., m$$

SVM also has a non-linear form which uses kernel trick [11]. It is similar to the original linear SVM except that the dot products are replaced by non-linear kernel functions such as:

$$k(x, x_i) = (\Phi(x_i), \Phi(x))$$

By doing so the maximum-margin hyperplane is fit into a transformed feature space which can be non-linear. By using different kernel functions, the SVM algorithm can construct a variety of learning machines. Commonly used kernels include:

- Polynomial classifiers of degree d: $k(x, x_i) = (k(x, x_i) + \Theta)^d$
- Gaussian Radial Basis Function: $k(x, x_i) = \exp(-\frac{\|x - x_i\|^2}{2\sigma^2})$
- Sigmoid: $k(x, x_i) = tanh(k \cdot (x \cdot x_i) + \Theta)$

In submitted runs with relevance feedback, all experiments were done 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 done 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 relevance feedback. The method for combining multiple descriptors in SVM was introduced in [12].

5 Rushes Task

Having a relatively opened task definition, the rushes task was an interesting challenge for our team. Here, we tried to generate an effective tool for manipulating this specific type of data - unorganised, repetitive, yet essential for video production. Having previous experiences with large collections of rushes [13] the team from University of Bristol proposed a video summarisation system targeting intuitive browsing of large video archives [14]. Initially, the camera work classification module detects and annotates regions with appropriate camera motion. An arousal value determined using affective modelling is assigned to each extracted key-frame and this value is used to optimally lay out the final video summary on a single display or page.

5.1 Camera work

In order to divide the video into consecutive segments of camera motion, we extended the previous work [15] used for the camera motion task in TRECVID 2005. First the shot boundary detector is applied and then the shots are subdivided into segments of camera motion.

The objective here is to translate the affine 6 parameter global motion model estimated from P-frame motion compensation vectors into physical motion, interpretable by humans, such as pan, tilt, or zoom. To do this, the global motion model is reformulated as:

$$\begin{pmatrix} dx_i \\ dy_i \end{pmatrix} = \begin{pmatrix} pan \\ tilt \end{pmatrix} + \begin{pmatrix} zoom \cdot x - rot \cdot y + hyp1 \cdot x + hyp2 \cdot y \\ zoom \cdot y + rot \cdot x - hyp1 \cdot y + hyp2 \cdot x \end{pmatrix} \tag{13}$$

Then two statistical hypotheses are tested on each parameter. H_0 assumes that the parameter is significant, the second one H_1 assumes that the component is not significant, i.e. equals zero. The likelihood function for each hypothesis is defined with respect to the residuals between the estimated model and the MPEG motion vectors. These residuals are supposed to follow the bi-variate Gaussian law. The decision on the significance is made by a comparison of the log-likelihood ratio with a threshold.

As the BBC rushes for TRECVID 2006 contain a lot of small camera motions not relevant for the segmentation, the significance values are learned during a fixed number of frames before testing if a change in one of the motion parameters appears which determines the end of a motion segment. This forces the motion segments to be of a minimal size. In addition, segments with a too small motion amplitude are detected and automatically classified as a static camera. Then, the remaining segments are classified based on the mean values of the significance as pan left/right, tilt up/down, zoom in/out, rotation, sideways travelling up/down left/right, zoom in/out + rotation, complex motion or static camera.

5.2 Affective modelling

The processing of the rushes for enabling non-linear content access starts with applying a newly developed methodology for modelling the "experience" of the recorded data.

We approach the modelling of the "experience" of the rushes by extending our previous work on arousal modelling [16]. 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 video, we model the arousal time curve that represents the variations in experience from one time stamp to another. In the context of rushes analysis, high arousal values ideally represent the parts of the recorded data with high excitement, as compared to more-or-less monotonous parts represented by low arousal values. The obtained curve can be used to estimate the parts in the data that might be relevant to the editor in the first place.

The system firstly extracts the fundamental audiovisual features from the rushes data that are related to the arousal. These features are sound energy level, zero crossing rate, pitch value and motion intensity. Then the extracted features are nonlinearly transformed so that the abrupt changes in the feature values are emphasized and the relatively smooth regions are suppressed. Finally, a linear combination of these values are smoothed to get the final excitement curve of the whole video data.

Figure 5: News sequence 20051121_125800_CNN_LIVEFROM_ENG.mpg from the TRECVID 2006 search corpus, summarised using layout parameters $\mathcal{N} = 70$ and $\mathcal{R} = 3/5$. Repetitive content is always presented by the smallest frames in the layout. On the other hand, outliers are presented as big (e.g. a commercial break within a newscast, row 2 frame 11) which is very helpful for the user to swiftly uncover the structure of the presented sequence.

5.3 Layout

In order to present a large collection of key-frames extracted from the rushes in an efficient and effortless way, we follow the narrative grammar of comics, and using its universal and intuitive rules, we lay out visual summaries in an efficient and user centered way. The constraint of spatial layout dependance on time flow is introduced, where the time flow of video sequence is reflected by ordering the frames in left-to-right and top-to-bottom fashion. Excluding this rule would impede the browsing process. Given the requirement that aspect ratio of key-frames in the final layout has to be the same as aspect ratio of the source video frames, the number of possible spatial combinations of frame layouts will be restricted and the frame size ratios have to be rational numbers (e.g. 1:2, 1:3, 2:3). The final layout is created using an discrete optimisation algorithm [14]. This is an sub-optimal algorithm that utilises dynamic programming (DP) to find the best solution in very short time. Results presented in [14] show that the error introduced by the sub-optimal model can be disregarded. The layouts depicted in Figure 6 and Figure 5 show how the browsing of rushes as well as other type of the content can be fast and intuitive.

References

[1] 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.

[2] L. Primaux, J. Benois-Pineau, P. Krämer, and J.-P. Domenger. Shot boundary detection in the framework of rough indexing paradigm. In *TREC Video Retrieval Evaluation Online Proceedings, TRECVID'04*, 2004.

[3] B.S. Manjunath, J.R. Ohm, V.V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE trans. on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

Figure 6: A sequence from the TRECVID 2006 rushes corpus. Since there is a lot of repetition of the content, this type of data fully exploits functionality of the presented system: the largest frames represent the most frequent content and in some cases extreme outliers (e.g. a capture error due to an obstacle in row 1, frame 3); middle sized frames represent similar, but a bit different content to the group represented by the largest frames; the smallest frames are simple repetitions of the the content represented by the largest frames.

[4] S.L. Chiu. *Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification*. John Wiley and Sons, 1997.

[5] S. Deerwester, S.T.Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[6] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Latent semantic indexing for semantic content detection of video shots. In *International Conference on Multimedia and Expo (ICME)*, 2004.

[7] V Mezaris, H Doulaverakis, S Herrmann, Bart Lehane, Noel O'Connor, I Kompatsiaris, and M Strintzis. Combining textual and visual information processing for interactive video retrieval: Schema's participation in trecvid 2004. In *TRECVID 2004 - Text REtrieval Conference TRECVID Workshop*, MD, USA, 2004. National Institute of Standards and Technology.

[8] S. Robertson and K. Jones. Simple proven approaches to text retrieval. Tecnical report UCAM-CL-TR-356, ISSN 14762986, University of Cambridge, 1997.

[9] Yong Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8:644–655, 1998.

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

[11] B. Scholkopf. The kernel trick for distances. *Advances in Neural Information Processing Systems*, pages 301–307, 2001.

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

[13] J. Ćalić, N. Campbell, M. Mirmehdi, B. Thomas, R. Laborde, S. Porter, and N. Canagarajah. ICBR - multimedia management system for intelligent content based retrieval. In *International Conference on Image and Video Retrieval CIVR 2004*, pages 601–609. Springer LNCS 3115, July 2004.

[14] J. Ćalić and N. Campbell. Comic-like layout of video summaries. In *Proc. of the 7th Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2006)*, 2006.

[15] P. Krämer and J. Benois-Pineau. Camera motion detection in the rough indexing paradigm. In *TREC Video Retrieval Evaluation Online Proceedings, TRECVID'05*, 2005.

[16] A. Hanjalic and L.-Q. Xu. Affective video content representation and modeling. *IEEE Transactions on Multimedia*, 7:143–154, 2005.