K-Space at TRECVid 2006

Peter Wilkins, Tomasz Adamek, Paul Ferguson, Mark Hughes, Gareth J.F.Jones,
Gordon Keenan, Kevin McGuinness, Jovanka Malobabić, Noel E. O'Connor,
David Sadlier, Alan F. Smeaton
Centre for Digital Video Processing & Adaptive Information Cluster
Dublin City University (DCU), Ireland

Rachid Benmokhtar, Emilie Dumont, Benoit Huet and Bernard Merialdo
Département Communications Multimédia
Institut Eurécom
2229, route des Crêtes, 06904 Sophia-Antipolis, France

Evaggelos Spyrou, George Koumoulos and Yannis Avrithis
Image Video and Multimedia Laboratory National Technical University of Athens (ITI)
9 Iroon Polytechniou Str., 157 80 Athens, Greece

R. Moerzinger, P. Schallauer, W. Bailer
Institute of Information Systems and Information Management
Joanneum Research (JRS)
Steyrergasse 17, 8010 Graz, Austria

Qianni Zhang, Tomas Piatrik, Krishna Chandramouli and Ebroul Izquierdo
Department of Electronic Engineering
Queen Mary, University of London (QMUL), United Kingdom

Lutz Goldmann, Martin Haller, and Thomas Sikora
Technical University of Berlin, Department of Communication Systems (TUB)
EN 1, Einsteinufer 17, 10587 Berlin, Germany

Pavel Praks
Department of Information and Knowledge Engineering
Faculty of Informatics and Statistics, University of Economics, Prague (UEP)
W. Churchill sq. 4, 130 67 Prague 3, Czech Republic

Jana Urban, Xavier Hilaire, Joemon M. Jose
Department of Computing Science, University of Glasgow (UG)
University Avenue, Glasgow G12 8QQ, United Kingdom.

October 29, 2006

Abstract

In this paper we describe the K-Space participation in TRECVid 2006. K-Space participated in two tasks, high-level feature extraction and search. We present our approaches for each of these activities and provide a brief analysis of our results. Our high-level feature submission made use of support vector machines (SVMs) created with low-level MPEG-7 visual features, fused with specific concept detectors. Search submissions were both manual and automatic and made use of both low- and high-level features. In the high-level feature extraction

submission, four of our six runs achieved performance above the TRECVid median, whilst our search submission performed around the median. The K-Space team consisted of eight partner institutions from the EU-funded K-Space Network, and our submissions made use of tools and techniques from each partner. As such this paper will provide overviews of each partner's contributions and provide appropriate references for specific descriptions of individual components.

1 Overview of K-Space

K-Space is a European Network of Excellence (NoE) in semantic inference for semi-automatic annotation and retrieval of multimedia content [1]. K-Space is focused on the research and convergence of three themes: content-based multimedia analysis, knowledge extraction and semantic multimedia. Of the 14 European research institutions that comprise K-Space, 8 have participated as part of this K-Space TRECVid submission. This was our (K-Space) first year of TRECVid participation and we have plans for continued TRECVid engagement throughout the lifespan of the NoE, potentially with increased involvement from other K-Space partners who did not participate this year.

2 High-Level Feature Extraction

In this section we will present our work for the high-level feature extraction task. Our approach within this task was to take a very generic approach, that of training a Support Vector Machine (SVM) per feature, making use of the common TRECVid annotations and low-level MPEG-7 visual features, and combining this data with more specialised concept detectors (such as a face detector and a desert detector). The organization of this section is as follows, first we describe our generic SVM approach and second we describe each of the specialized concept detectors. Finally we will discuss our methods for using these outputs and the results we obtained.

2.1 Generic Support Vector Machine Approach (DCU)

For our generic approach to high-level feature extraction, we first examined the common TRECVid annotations to arrive at our training set annotations. Visual features were extracted from all NRKF keyframes in the training and test collections and through experimentation on the 2005 features, we tuned the various parameters and kernel functions from our SVM. In our experiments we used svm_light [2].

We extracted low-level visual features using several feature descriptors based on the MPEG-7 XM. These descriptors were implemented as part of the aceToolbox, a toolbox of low-level audio and visual analysis tools developed as part of our participation in the EU aceMedia project [3]. For the high-level feature extraction task we made use of six different visual descriptors. These descriptors were Colour Layout, Colour Moments, Statistical Texture, Homogenous Texture, Edge Histogram and Scalable Colour. A complete description of each of these descriptors can be found in [24].

The data from the low level features was converted to a format compatible with svm_light and normalised into the range -1 and 1. The SVM's were then trained and tested using different kernel functions including linear and polynomial, however it was the radial basis function (RBF) that performed the best for this task. Different parameters were optimised for this kernel, such as cost and the gamma parameter.

2.2 Motion Detection (JRS)

Camera motion can be used to infer higher level information, if combined with other analysis results or domain knowledge. For example, zooming on an object or person is an indicator of relevance, and in field sports, pans indicate the direction of the game. As visual grammar imposes constraints on the camera motion of sequences to be combined, it is an important search and selection criterion when searching for essence in order to re-use it in new productions.

The detection of camera motion in an image sequence addresses two basic problems. Firstly, the dominant motion in the sequence is not necessarily the camera motion, e.g. if a large object moves in front of a static camera, the dominant motion will be estimated as the object's motion. Secondly, different types of camera motion causing the same visual effect cannot be discriminated against, e.g. pan left and track left in cases where target is distant and amount of motion is small. Unlike other approaches which ignore the fact that camera motion can only be determined reliably over a larger time range and which accept the most dominant motion between a frame pair as the camera motion, our approach is to estimate a number of dominant motions. We assume that the camera motion is consistent and smooth over the time range and that it is the most dominant one (e.g. the one with the largest region of support).

The extraction algorithm is the same that was used for the TRECVid 2005 camera motion task from Joanneum Research [6]. It is based on feature tracking which is a compromise between spatially detailed motion description and performance. Feature trajectories are then clustered by similarity in terms of a motion model and the cluster representing the global motion is selected. The steps of the algorithm are as follows.

Feature tracking Feature tracking is done on the input image sequence using the Lucas-Kanade tracker, using an improved version of the OpenCV implementation.

Clustering of trajectories Instead of clustering feature displacements between pairs of frames, trajectories over a longer time window (0.3 to 0.5 seconds) are clustered to achieve a more stable cluster structure over time. The number of clusters is unknown in this problem, and not all trajectories exist throughout the whole time window. Clustering is done in terms of similarity to a four parameter motion model. The clustering algorithm is an iterative approach to estimating a motion parameter sequence for a set of trajectories and then re-assigning trajectories to the best matching parameter sequence.

Dominant cluster selection From the clusters resulting from the clustering step, the one representing the dominant motion of the sequence is selected. This decision is done over a longer time window (up to several seconds), based on the size of the cluster (i.e. the number of features which are subject to this motion) and its temporal stability.

Camera motion detection The camera motion detection step analyzes the motion parameter sequence which has been found to represent the dominant motion and detects the presence of pan, zoom and tilt. The detection is done in a time window, for which the accumulated x- and y translation and the multiplied scale factor are calculated. In order to be robust against short time motion, the input is median filtered.

The description of the camera motion analysis is in MPEG-7 format, more specifically the camera motion descriptors are attached to the visual shots using the MPEG-7 CameraMotion descriptor (MPEG-7 Part 3, Visual [23]) compliant to the Detailed AudioVisual Profile (DAVP) from JOANNEUM RESEARCH [5]. For one or more segments per shot, the following types of motion are described: pan left/right, tilt up/down, roll CW/CCW, zoom in/out and static.

2.3 Face statistics (TUB)

The goal of this module is to extract statistics describing visible faces within a shot.

Initially a very robust component-based face detection approach proposed by Goldmann et al. [15] was used. Although it yields much better detection performance for high resolution (PAL) images than the widely used holistic approach by Viola & Jones [37], it did not work reliably for the subsampled low resolution (CIF) images of the TRECVid 2005 and 2006 datasets. Thus, the latter approach with the extensions proposed by Lienhart et al. [20] was finally adopted.

Image regions are described using binary Haar-like features that can be efficiently computed using an integral image. While Viola et al. [37] used only vertical and horizontal feature prototypes, Lienhart et al. [20] considered an extended set by adding rotated and surrounding feature prototypes. Applying the final 14 feature prototypes to an image region leads to a large overcomplete set of features. A supervised learning approach based on a classifier cascade is utilized for learning the face patterns based on these features from given training images. A weak classifier consists of a single feature, a corresponding threshold and a parity and achieves only a very low performance individually. A strong classifier is built by combining multiple weak classifiers using weighted summation and a thresholding operation. A feature selection strategy based on Adaboost is used to select a small subset of suitable weak classifiers. In order to achieve both low error rates and a low computational complexity, a cascade of strong classifiers with low complexity is used instead of a monolithic classifier with a very high complexity. The detector was trained for frontal faces only.

In order to derive face statistics on shot-level, two different strategies were used: In approach 1 the face detector is applied only to the keyframe of each shot while in approach 2 it is applied to each frame of a shot and a region based tracking approach is used to establish temporal correspondences. Since strategy 2 was too slow to process the whole TRECVid dataset within the given time frame, strategy 1 was finally used.

The final face statistics were derived by counting the number of faces within a shot and calculating the normalized size of the largest face with respect to the image dimensions. These statistics were exported in an extended MPEG-7 description scheme provided by JRS.

2.4 Outdoor Detection (Eurécom)

The system used for the detection of ourdoor shots tasks is functionally very similar to that used in TRECVid 2005 by Eurécom [17], but with a different method for combining classifiers. This year, we pursued our research on fusion of classifier outputs aimed at high-level feature extraction. We used color and texture features extracted from image regions located both around salient point and around homogeneous image patches, these features are then introduced in separate SVM classification systems (one per feature type as described in [33]) trained on the *outdoor* concept using the first half of the development data set. The fusion of classifiers outputs is finally provided by training a multi-layer perceptron neural network [8] on the second half of the training data. More details about this entire framework and its performance can be found in the notebook paper [7].

2.5 Specific Concept Detectors (ITI)

This section summarizes the approach followed for the extraction of certain semantic concepts in TRECVid video sequences. More specifically, the following procedure aims to detect 7 high-level features: *desert*, *vegetation*, *mountain*, *road*, *sky*, *fire-explosion* and *snow*, using the extracted keyframes from a video sequence.

For the representation of the low-level color and texture features in a given keyframe, a description based on the MPEG-7 Dominant Color Descriptor (DCD) and the MPEG-7 Homogeneous Texture Descriptor (HTD)

[22] has been selected. The k-means clustering method is applied on the RGB values of the keyframe, dividing it in k regions. The centroids of these regions are actually the dominant colors. The texture properties are described by the HTDs, one for each region of the image. All the visual descriptions of the keyframe are then scaled and merged into a unique vector.

Clustering is performed on all the descriptions of the training set with the *subtractive clustering* [11] method. This way, both the number of the clusters and their corresponding centroids are estimated. Each cluster may or may not represent a high-level feature and each high-level feature may be represented by one or more clusters. For example, the concept *desert* can have more than one instances differing in i.e. the color of the sand, each represented by the centroid of a cluster. Moreover, in a cluster that may contain instances from the semantic entity i.e. *sky*, these instances could be mixed up with parts from i.e. *sea*, if present in an image.

A "Region Thesaurus" that contains all the "Region Types" that are encountered in the training set is then constructed. These region types are the centroids of the clusters and all the other feature vectors of a cluster are their "synonyms". The use of the thesaurus is to facilitate the association of the low-level features of the image with the high-level concepts. Principal component analysis (PCA) is then applied in order to reduce the dimensionality and facilitate both training and performance of the high-level feature detectors.

After the construction of the region thesaurus, a "model vector" is formed for each keyframe. Its dimensionality is equal to the number of concepts that constitute the thesaurus. The distance of a region-to-region type is calculated as a linear combination of the DCD and HTD distances, respectively. The MPEG-7 standardized distance is used for the HTD and Euclidean distance is used for the DCD. A linear combination is then used to fuse the distances as in [34]. 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 1.

$$D_m[min\{d_i^1\}, min\{d_i^2\}, ..., min\{d_i^{N_C}\}], i = 1, 2, 3, 4 \quad (1)$$

For each of the 7 semantic concepts mentioned earlier in this subsection, a separate neural network (NN) is trained. The input of the NN is the model vector and the output represents the distance of each region to the corresponding semantic concept.

2.6 Specific Concept Detectors (QMUL)

In the feature extraction task, QMUL contributed with the extraction of the following four features: "US-Flag", "Boat/Ship", "Weather" and "Maps". These features were extracted by a two-stage framework. The first stage uses a high-recall, moderate-precision classifier which is trained to obtain a subset of shots relevant to the semantic feature. The second stage uses a high-precision classifier which is trained and applied on the subset obtained by the first module, in order to filter out false alarms. The framework is designed to handle the very large TRECVid dataset, considering both the classifier performance and the processing time.

The framework includes three modules developed within the MMV group in QMUL: text based latent semantic analysis (LSA) for image classification; a particle swarm optimisation based image classifier; and an ant colony based image classifier. Among these modules, text based and particle swarm optimisation (PSO) based image classifiers were used as first stage classification modules, while the ant colony based image classifier was used as a second stage classifier.

A brief introduction to each of the three modules follows:

Latent Semantic Analysis (LSA) is able to extract and infer relations on expected contextual usage for words (terms) in textual data [13]. In our text-based video retrieval module, the first step is to represent textual data as a term-document matrix. This step includes word stemming, stopword removal according to a well-defined stopword list, and finally term-document concurrence frequency counting and normalisation. In the next step a singular value decomposition (SVD) transformation is performed on the defined matrix. SVD is a dimensionality reduction technique which provides reduced-dimension approximations to both the column space and the row space of the Vector Space Model.

The next module is the ant colony based image classifier where the ant colony optimisation (ACO) and its learning mechanism is integrated with the COP-k-means to address image classification problem [25]. The COP-k-means is a semi-supervised variant of k-means, where initial background knowledge is provided in the form of constraints between instances in the dataset. The integration of ACO with a COP-k-means makes the classification process less dependent on the initial parameters, so that it becomes more stable.

Particle swarm pptimisation (PSO) 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 [10]. To improve the performance of the classification algorithm, fuzzy inference rules are constructed along with binary particle swarming to merge the classification results from multiple MPEG - 7 descriptors [9]. The rules were explicitly weighted based on the ability of the descriptor to classify different features/concepts.

For the search task, general topics were selected from TRECVid 2005 topics and the classification result was used as an inter-video semantic feature biasing filter.

2.7 Building & Crowd Detection (DCU)

Our building detection work was adopted from techniques developed to detect buildings in a corpus of personal digital photographs. The following description is taken from [21].

We adopt a multi-scale approach that relies on edge detection to extract an edge orientation-based feature description of the image, and apply an SVM learning technique to infer the presence of a dominant building object. Earlier testing of this approach on a collection of digital photographs exploited prior knowledge on the image context through an assumption that all input images are outdoor, i.e. indoor/outdoor classification (the context determination stage) has been performed. This information was not available for the TRECVid collection, however we still ran our approach on the TRECVid collection without this information. Whilst a more formal evaluation of the success of this approach is required, an initial results examination revealed that performance degradation was not great.

Our crowd detection technique was taken from our work in discovering events in field sports. The following description is taken from [31].

It is proposed that crowd image detection may be performed by exploiting the inherent characteristic that, in the context of a typically non-complex image environment, such images are relatively detailed. It is proposed that discrimination between detailed and non-detailed pixel blocks may be made by examining the number of non-zero frequency (AC) Discrete Cosine Transform (DCT) coefficients used to represent the data in the frequency domain. It may be assumed that an (8x8) pixel block, which is represented by very few AC-DCT uniform coefficients, contains spatially consistent, non-detailed data. Whereas, a block which requires a considerable amount of AC-DCT coefficients for it's representation, may be assumed to consist of relatively more detailed information.

In field-sports video content, the majority of images capture relatively sizeable monochromatic, homogeneous regions e.g. grassy pitch or a player's shirt. Therefore, in the context of this limited environment, it is proposed that crowd images may be isolated by simply detecting such uniformly, very high frequency images. Each I-frame is divided into four quadrants. For each quadrant of each image, the AC-DCT coefficients of every (8x8) luminance pixel block are analysed. If the number of coefficients used to encode such blocks is greater than a pre-selected threshold, it can be deduced that the block represents reasonably complex data, and is counted - obtaining an overall value representing the number of high frequency blocks, per total number of blocks, for each quadrant. Values for both mean number of high-frequency blocks (HF_{mean}) and standard deviation per quadrant (σ_{qx}), are calculated from the four quadrant values. It was noted that for uniform crowd images, HF_{mean} and σ_{qx} should have high and low values respectively. A crowd image confidence feature set, $\{Fv3\}$, is calculated as follows:

$$\{Fv3\} = HF_{mean} - Avg(\sigma_{q1}, \sigma_{q2}, \sigma_{q3}, \sigma_{q4}) \qquad (2)$$

Further information on our crowd detection, and more generally our event detection in field sports can be found in [30].

2.8 Fusion of detector outputs

Of our six submissions to feature detection in TRECVid 2006, three were submissions which used the fusion of the outputs of other runs. Two of these runs made use of Dempster-Shafer combination of evidence framework, whereas the third utilized our work on automatic weight generation for fusion [38].

The Dempster-Shafer submissions combined our baseline SVM data with several of the specialized concept detectors mentioned earlier. For this combination we required parameters which specified degrees of belief that a particular feature was performing well. We obtained these parameters through experimentation on the training collection. For specific details of the Dempster-Shafer combination framework refer to [12, 32, 26, 18].

Our automatic weight generation work was initially designed for the query-time fusion of multiple result lists for retrieval tasks. However we can apply these techniques to the fusion task for features. For this submission we fuse together the predictions of the baseline SVM with the predictions of the High-Level SVM. A brief description of the actual weight generation and fusion process used for this submission can be found in Section 3.3.2.

2.9 Results

We submitted six runs for our high-level feature extraction submission. Those six runs were:

Baseline ***(A_KSpace-base_6)*** The predications of the low-level visual SVM trained using the common TRECVid annotations.

Best-Breed ***(A_KSpace-bb_5)*** The specific concept detectors, where there was no specific concept detector the output from the baseline was used.

DS 1 ***(A_KSpace-DS1_2)*** A combination of the baseline with specific concept detectors using Dempster-Shafer, with parameters determined by experiments on the training data.

DS 2 ***(A_KSpace-DS2_1)*** As above with an alternate set of parameters and concepts used.

HighLevelSVM ***(A_KSpacehighSvm_4)*** A SVM built on the outputs of the specific concept detectors and the output of the baseline SVM, using the common TRECVid annotations.

FusedSVM ***(A_KSpaceSC_3)*** A fusion of the outputs of the baseline SVM and the high-level SVM.

Our results are shown in table 1 and are shown in comparison to the TRECVid median for this year.

We can derive a few things from our result. Firstly that our best result was our baseline submission which was better than median in 17 out of the 20 evaluated features when compared by inferred average precision. Those features in which we performed poorly correlated to poor median performance.

Of our remaining runs all of the fusion runs had a majority of features performing above median. Furthermore each of our fusion runs had features for which it outscored the baseline, lending support to the need for further exploration of these fusion strategies.

3 Search

In this section, we will present our work for the search task for TRECVid 2006. For this task we participated in both manual and fully automatic search. Our search systems made use of low-level visual features, ASR transcripts and the outputs of our high-level feature extraction task. We also had available further content analysis techniques (such as audio classification, and Latent Semantic Indexing of images) as inputs into our search systems. The rest of this section is organized as follows. Firstly we will describe the additional content analysis that was performed for the search task. Second we will present our manual search system, followed by our automatic search system. Finally we will present our results for the search task.

We introduce now our work on Latent Semantic Indexing for image retrieval, which was used for pseudo-relevance feedback in our manual submissions, and our audio classification which was used to compliment our ASR retrieval by boosting those shots which contained some form of speech.

3.1 Latent Semantic Indexing for automated intelligent image retrieval (UEP)

Numerical linear algebra is used as a basis for information retrieval in the retrieval strategy called Latent Semantic Indexing (LSI) [16]. LSI can be viewed as a variant of a vector space model, where the database is represented by the document matrix, and a user's query is represented by a vector. LSI retrieval is based upon a low-rank approximation of the original document matrix via singular value decomposition (SVD) or other numerical methods. The numerical methods are used as an automatic tool for identification and removing redundant information and noise from data. The next step of LSI retrieval involves the computation of the similarity coefficients between the filtered user's query and filtered document matrix. The well-known Cosine similarity can be used as a similarity measure.

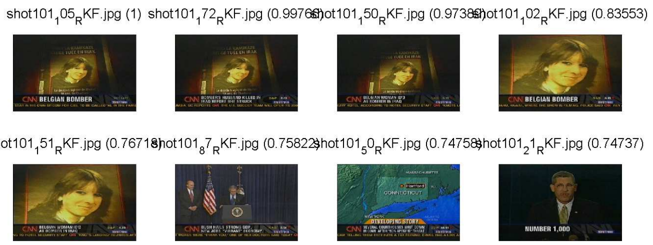

shot101$_1$05$_R$KF.jpg (1)
shot101$_1$72$_R$KF.jpg (0.99766)
shot101$_1$50$_R$KF.jpg (0.97389)
shot101$_1$02$_R$KF.jpg (0.83553)
BELGIAN BOMBER
BOMBER'S HUSBAND KILLED IN IRAQ BEFORE SHE STRUCK
BELGIAN WOMAN ID'D AS BOMBER IN IRAQ
BELGIAN BOMBER
shot101$_1$51$_R$KF.jpg (0.76718)
shot101$_8$7$_R$KF.jpg (0.75822)
shot101$_5$0$_R$KF.jpg (0.74758)
shot101$_2$1$_R$KF.jpg (0.74737)
BELGIAN WOMAN ID'D AS BOMBER IN IRAQ
BUSH HAILS STRONG GDP, NEW JOBS, "VIBRANT" ECONOMY
CONNECTICUT
DEVELOPING STORY
SEVERAL COURTHOUSES SHUT DOWN IN CONN. AFTER "NON-SPECIFIC" THREAT
NUMBER 1,000

Figure 1: An example of the SVD-free LSI keyframe similarity user-interface. The query image (shot101_105_RKF.jpg) is in the left upper corner and has a similarity coefficient of 1. All of the 4 most similar images are related to the same topic. I

Originally, LSI was developed for the semantic analysis of a large amount of text documents. We extended the original LSI for intelligent image retrieval [27]. In our approach [27, 28], a raster image is coded as a sequence of pixels. Then the coded image can be understood as a vector of a m-dimensional space, where m denotes the number of pixels (attributes). Let a symbol A denote an $m \times n$ term-document matrix related to m keywords (pixels) in n documents (images). Let us remind that the (i, j)-element of the term-document matrix A represents the colour of i-th position in the j-th image document [27, 28]. We also showed that image retrieval can be powered very effectively when the time consuming Singular Value Decomposition of the original LSI is replaced by the partial symmetric eigenproblem which can be solved very effectively by using fast iterative solvers [28]. We have successfully used this approach especially for surveillance in hard industry [29], web image classification [19] and as an automated tool for the large-scale iris recognition problem [28], prior to its use in the K-Space participation in TRECVid 2006.

For TRECVid 2006 we processed each video of the test collection separately by developed SVD-free LSI approach, see Figure 2. This meant that we created 259 separate document matrices. Although the document matrix of each task required several hundered Megabytes of RAM, all computations were stable and fast on a Pentium4 PC with 3 GHz CPU and 2 GB RAM. One of the reasons for this is that singular values of TRECVid 2006 keyframes tend to decrease quite fast so that only 8 extremal eigenvalues and corresponding eigenvectors of the large partial symmetric eigenproblem were computed and stored in memory in all cases. The second reason for the fast execution is that we used an efficient implementation of linear algebra algorithms which assume several key implementation details [28]. Finally, the keyframe similarity task of each directory required only seconds, as shown in Table 2.

Feature	Median	Baseline	Best-Breed	DS1	DS2	HighLevelSVM	FusedSVM
sports	0.254	0.3454	0.1085	0.3298	0.3298	0.2879	0.3381
weather	0.253	0.2004	0.0078	0.2018	0.2018	0.1749	0.1985
office	0.004	0.0045	0.0045	0.0045	0.0045	0.0012	0.0028
meeting	0.111	0.1788	0.1788	0.0277	0.0277	0.1171	0.1706
desert	0.021	0.0588	0.0002	0.0567	0.0128	0.0015	0.0352
mountain	0.038	0.0546	0.0002	0.0357	0.0584	0.0119	0.0393
waterscape	0.039	0.1361	0.1361	0.1361	0.1361	0.0806	0.1251
corporate-leader	0.001	0.0068	0.0068	0.012	0.012	0.0313	0.0175
police	0.007	0.0146	0.0146	0.0146	0.0146	0.0104	0.0154
military	0.049	0.0773	0.0157	0.0636	0.0636	0.0492	0.0696
animal	0.004	0.0042	0.0042	0.0041	0.0041	0.0003	0.0043
computer_tv_screen	0.114	0.2716	0.2716	0.0237	0.0237	0.1417	0.2609
flag-us	0.078	0.1948	0.073	0.1734	0.1734	0.0043	0.1531
airplane	0.011	0.0105	0.0105	0.0129	0.0129	0.0047	0.0201
car	0.079	0.19	0.19	0.1699	0.1699	0.0785	0.1526
truck	0.019	0.045	0.045	0.0419	0.0419	0.0028	0.0253
people-marching	0.02	0.0282	0.0282	0.0026	0.0026	0.0222	0.0381
explosion	0.025	0.0679	0.0008	0.0734	0.0734	0.0029	0.0029
maps	0.17	0.2484	0.0003	0.2432	0.2432	0.1196	0.2437
charts	0.062	0.0702	0.0702	0.0702	0.0702	0.0004	0.0403
No. Higher than median	-	17	10	15	15	8	16

Table 1: 2006 K-Space Feature Results

Properties of the document matrix *A*	
Number of keywords:	352×240 = 84 480
Number of documents:	227
Size in memory:	146.3 MB
The SVD-Free LSI processing parameters	
Dim. of the original space	227
Dim. of the reduced space (k)	8
Time for A^TA operation	1.375 secs.
Results of the eigensolver	0.047 secs.
The total time	1.422 secs.

Table 2: Image retrieval using the SVD-free Latent Semantic Indexing method related to the 20051202_125800_CNN_LIVEFROM_ENG directory; Properties of the document matrix (up) and LSI processing parameters (down). Decompressing of original JPGs onto bitmaps required 3.938 secs.

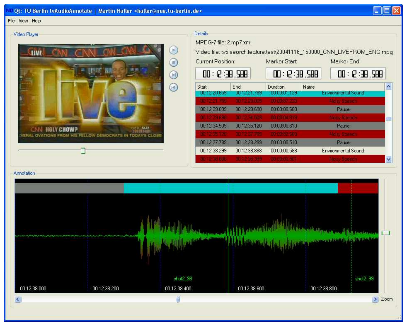
3.2 Audio classification/segmentation (TUB)

Audio classification/segmentation identifies the nature of an audio signal for a given closed set of categories and provides homogeneous temporal segments. Here, the following six categories were used: pause, clean speech, noisy speech, pure music, music and speech as well as environmental sound.

For the audio analysis process, the audio stream of the TRECVid videos is mixed down to a mono audio signal with a sample rate of 22050 Hz. After that, feature extraction determines 13 mel cepstral frequency coefficients (MFCCs) for each analysis frame with a 20 ms duration and a 10 ms hop size. The mel filter bank consists of 30 mel-warped triangular overlapped band pass filter between 64 Hz and 11025 Hz. A Gaussian mixture model (GMM) with 32 mixtures is trained for each category. These models are used for the maximum likelihood classification of sub-segments with duration of 0.5 seconds. Subsequently, sub-segments with the same recognized category are merged into one segment. In the end, the audio classification/segmentation provides the begin/end time as well as confidence values for all categories for each segment. An extended version of the MPEG-7 ClassificationType descriptor used in combination with the AudioSegment descriptor is used for storage and exchange of these results. The non-standard extension enables the assignment of multiple classes to one segment along with optional confidence values.

The ground truth for the six categories was created from 10 selected videos of the TRECVid 2006 training set. The total duration of annotated segments is 5 hours and 50 minutes. A 70 % / 30 % training/test data split is used for evaluation purposes. For this split, a classification experiment could achieve a recognition rate of 75.86 % for each segment. Even if this result is not highly accurate, the audio segmentation results for the whole set of TRECVid 2006 videos are nevertheless useful information for further content analysis or fusion techniques for video retrieval.

For the time-consuming annotation task, the audio segmentation program "tvAudioAnnotate" (Fig. 2) was created and used by TUB. In addition to the playback, visualization, and segmentation capabilities for the audio stream of MPEG video files, the program provides also a synchronized playback of the visual stream and the visualization of TRECVid reference shot boundaries. This audiovisual support during annotation shall increase the correctness of manual segmentation.



MPEG-7 file: 2.mp7.xml
Video file: tv5.search.feature.test\20041116_150000_CNN_LIVEFROM_ENG.mpg

Current Position: 00:12:38.588 Marker Start: 00:12:38.588 Marker End: 00:12:38.588

Start	End	Duration	Name
00:12:20.659	00:12:21.789	00:00:01.129	Environmental Sound
00:12:21.789	00:12:29.009	00:00:07.220	Noisy Speech
00:12:29.009	00:12:29.690	00:00:00.680	Pause
00:12:29.690	00:12:34.509	00:00:04.819	Noisy Speech
00:12:34.509	00:12:35.120	00:00:00.610	Pause
00:12:35.120	00:12:37.789	00:00:02.669	Noisy Speech
00:12:37.789	00:12:38.299	00:00:00.510	Pause
00:12:38.299	00:12:38.888	00:00:00.588	Environmental Sound
00:12:38.888	00:12:39.389	00:00:00.501	Noisy Speech

Figure 2: TRECVid audio annotation software: tvAudioAnnotate

We now present two variants of our retrieval systems, a manual and an automatic system.

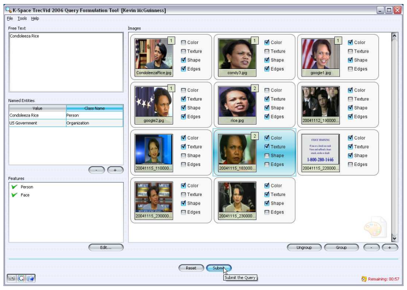
3.3 Manual Retrieval

Manual retrieval was performed by DCU and our system was divided into two parts, a query formulation tool for the user to create queries from topic descriptions, and an automatic retrieval system which processed queries to create the final result set.

3.3.1 Query Formulation Tool

The query formulation tool allows a user to select a range of query options for a given topic. The query options available to the user are:

- Add query images and for each query image the user can select which visual features to use (such as colour, edges).
- Group query images into visually similar clusters.
- Enter a free text query.
- Select high-level semantic features to use for a query, and to select whether each should have a positive or negative impact (e.g. for a query for “cars” we might use a negative “face” filter).



Free Text: Condoleeza Rice

Value	Class Name
Condoleeza Rice	Person
US Government	Organization

Features: ✓ Person ✓ Face

CondoleezaRice.jpg [1]: □ Color □ Texture ☑ Shape ☑ Edges
condy3.jpg [1]: ☑ Color □ Texture ☑ Shape ☑ Edges
google1.jpg [1]: ☑ Color □ Texture ☑ Shape □ Edges
google2.jpg [1]: □ Color ☑ Texture ☑ Shape ☑ Edges
rice.jpg [2]: □ Color ☑ Texture ☑ Shape ☑ Edges
20041112_190000...: ☑ Color □ Texture ☑ Shape ☑ Edges
20041115_110000...: ☑ Color ☑ Texture □ Shape □ Edges
20041115_183000... [2]: ☑ Color ☑ Texture □ Shape □ Edges
20041115_220000...: ☑ Color ☑ Texture ☑ Shape □ Edges
20041115_230000...: ☑ Color □ Texture ☑ Shape □ Edges
20041115_230000...: ☑ Color ☑ Texture ☑ Shape □ Edges

Figure 3: TRECVid Query Formulation Tool

Name	AP	P@5	P@10	Recall
Text	0.1611	0.6	0.3	85%
Edge	**0.3214**	0.6	0.4	66%
Colour Layout	0.0154	0.0	0.0	40%
Colour Struct.	0.0032	0.0	0.0	22%

Table 3: 0135 Feature Results

Figure 3 shows a screenshot of the user interface. The single expert user who formulated all 24 queries from the topic descrptions was allowed up to 15 minutes for query generation per topic. During this time the user could modify the query but received no feedback during this time as to how this query might perform.

For our manual experiments, we had one expert user conduct all 24 topics for this year’s search task. Before creating the manual queries for the 2006 topics our expert user was able to experiment with query performance by creating queries for the 2005 search task and received offline feedback as to how these queries performed. This is because the query formulation tool itself is unable to run queries or provide any feedback. Once formulated, the queries were fed into the retrieval system.

3.3.2 Retrieval Engine

The retrieval system used for our experiments is based upon our work for automatic weight generation [38], and a more thorough description of this system will appear in [39].

Our system generates query-time weights for the fusion of different information sources based upon the comparison of the score distribution differences of one information source as compared to another. This work is based upon our observations of information source performance for TRECVid retrieval queries, where an information source can be the output of a text search engine which has indexed the ASR, or low-level MPEG-7 visual features such as global colour, local colour or an edge histogram. When these features for a given topic are normalized and plotted, we observe that a correlation appears to exist between an information source whose top ranked documents undergo a rapid change in score and the information source which achieved the highest average precision for that topic. In other words, the best performing feature was generally the feature which exhibited this rapid change. This is demonstrated in Figure 4, with the performance figures for this graph shown in Table 3, where we can see that the greatest change in the top ranked shots is in the edge feature, and it is the edge feature which achieves the best average precision for this topic. For a complete description of these observations, and how we derive weights from them, refer to [38].

The retrieval engine for 2006 made use of low-level MPEG-7 visual features, ASR transcripts and high-level features. Our visual features were extracted from all

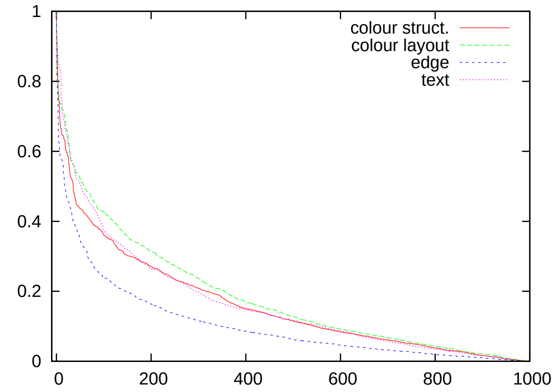


Figure 4: TRECVid topic 0135

RKF images and formed the basis of our visual index. Of the MPEG-7 visual descriptors available to us, we used an edge histogram descriptor, local colour descriptor, global colour descriptor and a homogenous texture descriptor. Full descriptions of these can be found in [24]. When we query a visual database, we rank the results utilizing a Euclidian distance metric.

We used the Zettair search engine [4] to provide text search capabilities for ASR retrieval. Because the ASR of a shot may not necessairly correspond to what is being shown visually by a particular keyframe, we employed a windowed weighting scheme whereby when a shot was found in the ASR we also returned the adjacent two shots, which are given decreased scores than the original.

The use of High-Level features in our system was to modify the final result list that was the result of the previous content-based retrieval. As such the introduction of High-Level features occurs at the end of the retrieval system. We employed a basic filtering approach for the application of these High-Level features. For each High-Level feature we determined a threshold for that feature which was used as a cut-off point for determining if a shot was successfully classified by that feature or not. This threshold was chosen through examination of the performance of the High-Level features in classifying the training collection. With this achieved we could then use the feature in a ‘positive’ or ‘negative’ manner.

If the High-Level feature was being used to give a ‘positive’ influence to the ranking, we first took the final ranking from the content-based search, and for each shot we performed a lookup for the candidate feature. If the shot being queried was above the threshold for that feature, then that score was given a boost to its score (typically a 10% increase of its current score). If the shot was not present above the threshold, then the score of the shot was not altered.

Conversely if the High-Level feature was used as a ‘negative’ influence, we again performed a lookup of every shot against that feature. If the shot did not appear above the threshold, then it received a boost to its score (the opposite of the ‘positive’ example). As such whilst our High-Level features were used as a final filtering step in the ranking, this filtering purpose was to subtlety alter the final ranking, rather than perform mass exclusions or changes to the ranking.

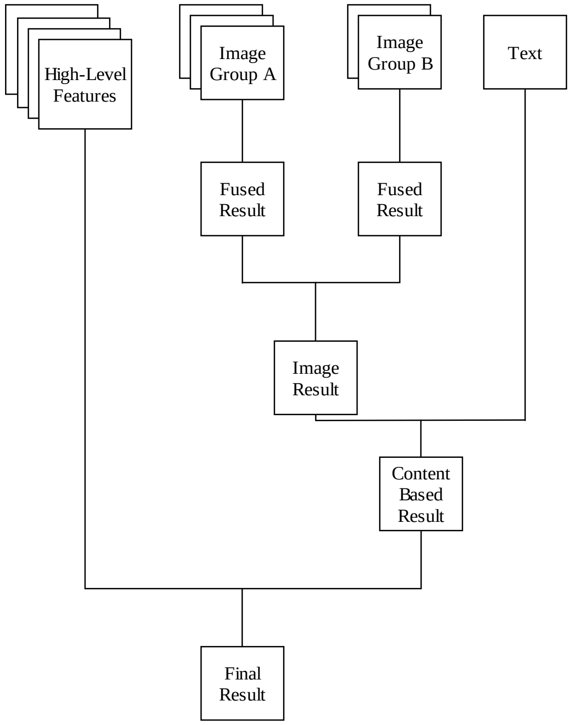


Figure 5: Retrieval fusion framework

Figure 5 illustrates our fusion framework for these experiments. Before we fuse any information sources together, we first normalize our source through MinMax normalization, formally shown in equation 3.

$$Norm_{score(x)} = \frac{Score_x - Score_{min}}{Score_{max} - Score_{min}} \qquad (3)$$

Once our sources have been normalized, we then generate our weights as described earlier in this section and in [38]. At this point we are left with weighted result lists for each information source, and we combine these lists by applying CombSUM [14]. Our exact order of operations for fusion is to first, for each query image, fuse together the outputs of a visual database search for that image (i.e. the results from a local colour query, edge histogram query etc.) such that we are left with one result list per image. Second, if the query images belong to an image grouping as identified by the user we will fuse together the results of each image that comprise that group into a single result. Third, all image group results will be fused together to form a single result for all visual queries. Fourth, we then fuse the results of the visual search with the ASR search. Finally we apply our High-level filters to modify the ranking of the final result list. As stated earlier, a more thorough explanation of this retrieval system can be found in [39].

In two of our runs we also applied, after the final combination, a pseudo-relevance feedback step. For each of the top 10 keyframe images in the final result list, we queried each against the LSI index (Section 3.1), and for the first five images found for each query we performed a lookup in the final result list and if the candidate image was found, its score was boosted by 10%.

3.4 Automatic Retrieval (UG)

Automatic retrieval experiments were conducted by the University of Glasgow, using their automatic retrieval system.

Two fully automatic runs were submitted to investigate the combination of various feature modalities (F_A_2_KSpace-F-2_2, F_A_2_KSpace-F-4_4). These runs are based on the same graph model, the ICG, as described in [36] and used for TRECVid 2006 runs by Glasgow University [35]. The graph is constructed using the terms from the textual index[1]. Furthermore the underlying visual features are the same as in [35]. In addition, peer information is employed in these runs based on the high-level feature submissions by the K-Space team.

The submitted results of run KSpace-1-DS_combo_plus-100 are the basis of the high-level features incorporated as peers in the ICG. Each of the 39 concepts is treated as a "peer group" in the ICG. Since the submitted results can also contain non-relevant shots per concept, only the first 100 shots are considered to belong to the corresponding peer group. All shots in one peer group are related (share a concept). Therefore, in the ICG a peer group is represented by a 100-clique (each shot in the group is linked to every other shot in the group).

In order to query the ICG, we need to choose a suitable restart vector before the random walk on the graph can be calculated. F_A_2_KSpace-F-2_2 is based on the textual topic description only (no query expansion), that is the restart vector is set to the terms extracted from the description field. F_A_2_KSpace-F-4_4 implements both query-by-keyword and query-by-example. In addition to the term nodes, the top 10 visual query results most similar to the given topic examples are chosen as the visual query nodes (see [35]).

3.5 Results

We submitted 6 runs as part of our search submission. These six runs were as follows:

M_A_2_KSpace_M_1 Manual run using only text and visual components. No High-Level features were used.

F_A_2_KSpace_A_2 Fully automatic run, as specified in 3.4.

M_A_2_KSpace_M_3 Manual run incorporating text, low-level visual information, motion information, and high-level feature data from our baseline feature run. Audio classification was used as a filter to boost shots which contained speech, and pseudo-relevance feedback was applied.

Run Name	MAP	Recall
Manual 1	0.031	0.13
Manual 3	0.035	0.15
Manual 5	0.031	0.14
Baseline	0.013	0.12
Auto 1	0.025	0.14
Auto 2	0.018	0.13

Table 4: 2006 Search Results

F_A_2_KSpace_A_4 Fully automatic run, as specified in 3.4.

M_A_2_KSpace_M_5 Manual run incorporating text, low-level visual information, motion information, and high-level feature data from our high-level SVM feature run. Audio classification was used as a filter to boost shots which contained speech, and pseudo-relevance feedback was applied.

M_A_1_KSpace_M_6 Baseline, text only run.

The results from these runs are presented in Table 4. The first observation that we can make is that our baseline run performs quite poorly. Further investigation into this will be required by an initial examination of our ASR index creation algorithms. This baseline is poor to begin with but it is encouraging to note that our other runs were able to build upon its performance. Our inclusion of high-level features did not have a significant impact upon precision, however it did seem to boost recall. Mechanisms will now need to be developed to see how this can be exploited into an increase into precision.

4 Conclusion

We have presented the K-Space participation in TRECVid 2006. This was our first participation in TRECVid and proved to be a very illumining experience, both in terms of the size of the task and the co-ordination effort in managing a very large group. Our results for the High-Level Feature Extraction task are good, whilst our search performance needs to be examined. Nevertheless our participation has been a positive experience for our partners and we look forward to greater participation in next years TRECVid activities.

5 Acknowledgments

The research leading to this paper was supported by the European Commission under contract FP6-027026 (K-Space).

[1] Unlike [35], the textual index is only expanded by 1 shot.

References

[1] KSpace Network of Excellence, information at http://www.k-space.eu/.

[2] svm_light, available from http://svmlight.joachims.org/.

[3] The AceMedia Project, available at http://www.acemedia.org.

[4] The Zettair search engine, available from http://www.seg.rmit.edu.au/zettair/.

[5] W. Bailer and P. Schallauer. The Detailed Audiovisual Profile: Enabling Interoperability between MPEG-7 based Systems. In *12^{th} International MultiMedia Modelling Conference (MMM'06)*, pages 217–224, Beijing, China, 2006.

[6] W. Bailer, P. Schallauer, and G. Thallinger. Joanneum Research at TRECVID 2005 – Camera Motion Detection. In *Proceedings of TRECVID Workshop*, pages 182–189, Gaithersburg, Md., USA, 11 2005. NIST.

[7] R. Benmokhtar, E. Dumont, B. Huet, and B. Merialdo. Eurécom at TRECVid 2006: Extraction of High-level Features and BBC Rushes Exploitation. In *TREC 2006, 15th Text Retrieval Conference, NIST, November 2006, Gaithersburg USA*, 2006.

[8] R. Benmokhtar and B. Huet. Classifier fusion : combination methods for semantic indexing in video content. In *ICANN 2006, International Conference on Artificial Neural Networks, 10-14 September 2006, Athens, Greece*, Sep 2006.

[9] K. Chandramouli, D. Djordjevic, and E. Izquierdo. Binary particle swarm and fuzzy inference for image classification. In *Proceedings of Proceedings of 3rd International Conference on Visual Information Engineering 2006*, pages 126–131, 1988.

[10] K. Chandramouli and E. Izquierdo. Image Classification using Self-Organising Feature Map and Particle Swarm Optimisation. In *Proceedings of 3rd International Conference on Visual Information Engineering 2006*, pages 313–316, 2006.

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

[12] A. P. Dempster. A generalization of the Bayesian inference. *Journal of Royal Statistical Society*, 30:205 – 447, 1968.

[13] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 281–285, 1988.

[14] E. A. Fox and J. A. Shaw. Combination of Multiple Searches. In *Proceedings of the 2nd Text REtrieval Conference*, 1994.

[15] L. Goldmann, U. Moenich, and T. Sikora. Robust face detection based on components and their topology. In *Electronic Imaging*, 2006.

[16] D. Grossman and O.Frieder. *Information retrieval: Algorithms and heuristics.* Kluwer Academic Publishers, Second edition, 2000.

[17] J. Jiten, F. Souvannavong, B. Merialdo, and B. Huet. Eurecom at TRECVid 2005: Extraction of High-level Features. In *TRECVid 2005, NIST, November 2005, Gaithersburg USA*, 2005.

[18] J. M. Jose, J. Furner, and D. J. Harper. Spatial Querying for Image Retrieval: A User Oriented Evaluation. In *ACM SIGIR*, pages 232 – 240, 1998.

[19] M. Labský, M. Vacura, and P. Praks. Web image classification for information extraction. In *First International Workshop on Representation and Analysis of Web Space (RAWS-05).* Prague, Czech Republic, http://ceur-ws.org/Vol-164/raws2005-paper7.pdf, September 2005.

[20] R. Lienhart, L. Liang, and A. Kuranov. An Extended Set Of Haar-Like Features For Rapid Object Detection. Technical report, Intel Research, 2002.

[21] J. Malobabić, H. LeBorgne, N. Murphy, and N. E. O'Connor. Detecting the presence of large buildings in natural images. In *CBMI 2005 - 4th International Workshop on Content-Based Multimedia Indexing*, 2005.

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

[23] MPEG-7. Multimedia Content Description Interface. Standard No. ISO/IEC n°15938, 2001.

[24] N. O'Connor, E. Cooke, H. le Borgne, M. Blighe, and T. Adamek. The AceToolbox: Low-Level Audiovisual Feature Extraction for Retrieval and Classification. In *2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, 2005.

[25] T. Piatrik and E. Izquierdo. Image classification using an ant colony optimization approach. In *Proceedings of 1st International Conference on Semantic and Digital Media Technologies*, 2006.

[26] V. Plachouras and I. Ounis. Dempster-Shafer Theory for a Query-Biased Combination of Evidence on the Web. *Information Retrieval*, 8(2):197–218, April 2005.

[27] P. Praks, J. Dvorský, and V. Snášel. Latent semantic indexing for image retrieval systems. In *SIAM Linear Algebra Proceedings, Philadelphia, USA*. International Linear Algebra Society (ILAS), http://www.siam.org/meetings/la03/proceedings/-Dvorsky.pdf, July 2003.

[28] P. Praks, L. Machala, and V. Snášel. *On SVD-free Latent Semantic Indexing for Iris Recognition of Large Databases*. Springer, In: V. A. Petrushin and L. Khan (Eds.) Multimedia Data mining and Knowledge Discovery (Part V, Chapter 24), 2006 (in print).

[29] P. Praks, J. Černohorský, V. Svátek, and M. Vacura. Human expert modelling using semantics-oriented video retrieval for surveillance in hard industry. In *ACM MobiMedia 2006: 2nd International Mobile Multimedia Communications Conference*. K-Space special session on Automatic Annotation and Retrieval of Multimedia Content, Alghero, Sardinia, Italy, September 2006.

[30] D. Sadlier and N. O'Connor. Event detection in field sports video using audio-visual features and a support vector machine. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(10):1225–1233, 2005.

[31] D. Sadlier, N. E. O'Connor, N. Murphy, and S. Marlow. A framework for event detection in field-sports video broadcasts based on svm generated audio-visual feature model. case-study:soccer video. In *IWSSIP'04 - International Workshop on Systems, Signals and Image Processing*, 2004.

[32] G. Shafer. A Mathematical Theory of Evidence. *Princeton University Press*, 1976.

[33] F. Souvannavong, B. Mérialdo, and B. Huet. Latent semantic indexing for semantic content detection of video shots. In *ICME 2004, IEEE International Conference on Multimedia and Expo, June 27th-30th, 2004, Taipei, Taiwan*, Jun 2004.

[34] E. Spyrou, H. vLeBorgne, T. Mailis, E. Cooke, Y. Avrithis, and N. O'Connor. Fusing MPEG-7 Visual Descriptors for Image Classification. In *International Conference on Artificial Neural Networks (ICANN)*, 2005.

[35] J. Urban, X. Hilaire, R. Villa, F. Hopfgartner, and J. M. Jose. Glasgow University at TRECVID. In *TRECVid 2006 – Text REtrieval Conference, TRECVID Workshop, Gaithersburg, Maryland, 13-14 November 2006*, 2006.

[36] J. Urban and J. M. Jose. Adaptive image retrieval using a graph model for semantic feature integration. In *Proc. of the 8th ACM SIGMM Int. Workshop on Multimedia Information Retrieval (MIR'06).* ACM Press, 2006.

[37] P. A. Viola and M. J. Jones. Robust real-time object detection. In *IEEE Workshop on Statistical and Computational Theories of Computer Vision*, 2001.

[38] P. Wilkins, P. Ferguson, and A. F. Smeaton. Using score distributions for querytime fusion in multimedia retrieval. In *MIR 2006 - 8th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2006.

[39] P. Wilkins, M. Koskela, T. Adamek, A. F. Smeaton, and N. E. O'Connor. TRECVid 2006 Experiments at Dublin City University. In *TRECVid 2006 – Text REtrieval Conference, TRECVID Workshop, Gaithersburg, Md., 13-14 November 2006*, 2006.