CLIPS-LSR Experiments at TRECVID 2006

Stéphane Ayache[1], *Jérôme Gensel*[2] *and Georges M. Quénot*[1]

[1] CLIPS-IMAG, BP53, 38041 Grenoble Cedex 9, France
[2] LSR-IMAG, BP53, 38041 Grenoble Cedex 9, France

Stephane.Ayache, Georges.Quenot@imag.fr

Abstract

This paper presents the systems used by CLIPS-IMAG and LSR-IMAG laboratories for their participation to TRECVID 2006 and the obtained results.

Shot boundary detection was performed using a system based on image difference with motion compensation and direct dissolve detection. This system gives control of the silence to noise ratio over a wide range of values and for an equal value of noise and silence (or recall and precision), the F1 value is 0.805 for all types of transitions, 0.833 for cuts and 0.727 for gradual transitions.

High level feature detection was performed using networks of SVM classifiers arranged in a variety of architectures and taking into account a variety of low level descriptors combining text, local and global information as well as conceptual context. The inferred average precision of our first run is 0.088.

The search system uses a user controlled combination of five mechanisms: keywords, similarity to example images, semantic categories, similarity to already identified positive images, and temporal closeness to already identified positive images. The mean average precision of the system (with the most experienced user) is 0.184.

1 Shot Boundary Detection

The CLIPS-IMAG team have participated to the Shot Boundary Detection (SBD) task with little modifications from previous participations. The system detects "cut" transitions by direct image comparison after motion compensation and "dissolve" transitions by comparing the norms of the first and second temporal derivatives of the images. It also contains a module for detecting photographic flashes and filtering them out as erroneous cuts and a module for detecting additional cuts via a motion peak detector. The precision versus recall or noise versus silence tradeoff is controlled by a global parameter that modifies in a coordinated manner the system internal thresholds. The system is organized according to a (software) dataflow approach and Figure 1 shows its architecture.

Very little modification was made relatively to the previous versions of the system, only minor adjustments of control parameter.

1.1 Cut detection by Image Comparison after Motion Compensation

This system was originally designed to evaluate the interest of using image comparison with motion compensation for video segmentation. It has been complemented afterward with a photographic flash detector and a dissolve detector.

1.1.1 Image Difference with Motion Compensation

Direct image difference is the simplest way for comparing two images and then to detect discontinuities (cuts) in video documents. Such difference however is very sensitive to intensity variation and to motion. This is why an image difference af-

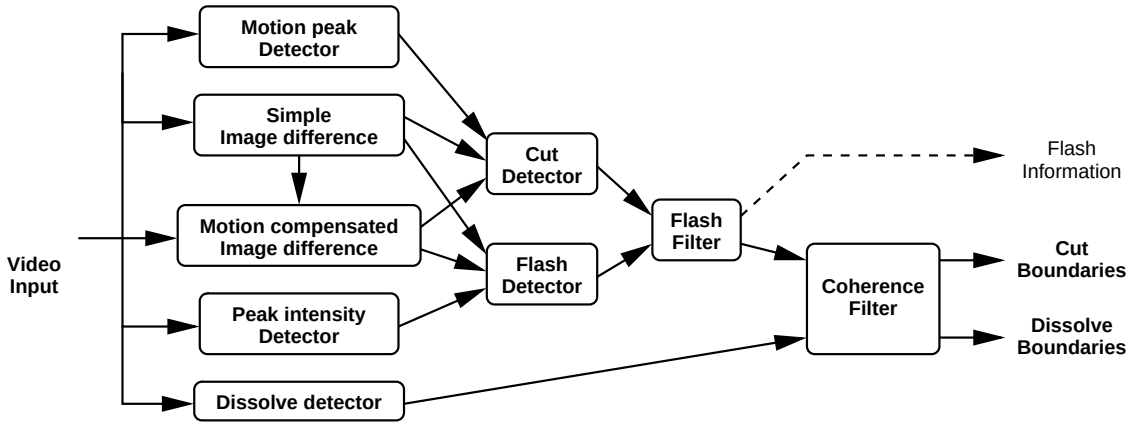

Figure 1: Shot boundary detection system architecture

ter motion compensation (and also gain and offset compensation) has been used here.

Motion compensation is performed using an optical flow technique [1] which is able to align both images over an intermediate one. This particular technique has the advantage to provide a high quality, dense, global and continuous matching between the images. Once the images have been optimally aligned, a global difference with gain and offset compensation is computed.

Since the image alignment computation is rather costly, it is actually computed only if the simple image difference with gain and offset compensation alone has a large enough value (i.e. only if there is significant motion within the scene). Also, in order to reduce the computation cost, the differences (with and without motion compensation) are computed on reduced size images (typically 88 $\times$ 60 for the NTSC video format). A possible cut is detected if both the direct and the motion compensated differences are above an adaptive threshold.

In order for the system to be able to find shot continuity despite photographic flashes, the direct and motion compensated image difference modules does not only compare consecutive frames but also, if needed, frames separated by one or two intermediate frames.

1.1.2 Photographic flash detection

A photographic flash detector module was implemented in the system since flashes are very frequent in TV news (for which this system was originally designed for) and they induce many false positives. Flash detection has also an interest apart from the segmentation problem since shots with high flash densities indicate a specific type of event which is an interesting semantic information.

The flash detection is based on an intensity peak detector which identify 1- or 2-frame long peaks on the average image intensity and a filter which uses this information as well as the output of the image difference computation modules. A 1- or 2-frame long flash is detected if there is a corresponding intensity peak and if the direct or motion compensated difference between the previous and following frames are below a given threshold. Flash information is used in the segmentation system for filtering the detected cut transitions.

1.1.3 Motion peak detection

It was observed from TREC-10 and other evaluations that the motion compensated image difference was generally a good indicator of a cut transition but, sometimes, the motion compensation was too good at compensating image differences (and even more when associated to a gain and offset compensation) and quite a few actual "cuts" were removed because the pre- and post-transition images were accidentally too close after

motion compensation. We found that it is possible not to remove most of them because such compensation usually requires compensation with a large and highly distorted motion which is not present in the previous and following image-to-image change. A cut detected from simple image difference is then removed if it is not confirmed by motion compensated image difference *unless* it also corresponds to a peak in motion intensity.

1.2 Dissolve detection

Dissolve effects are the only gradual transition effects detected by this system. The method is very simple: a dissolve effect is detected if the L_1 norm (Minkowski distance with exponent 1) of the first image derivative is large enough compared to the L_1 norm of the second image derivative (this checks that the pixel intensities roughly follows a linear but non constant function of the frame number). This is expected to detect dissolve effects between constant or slowly moving shots. This first criterion is computed in the neighborhood ($\pm$ 5 frames) of each frame and a filter is then applied (the effect must be detected or almost detected in several consecutive frames).

1.3 Output filtering

A final step enforces consistency between the output of the cut and dissolve detectors according to specific rules. For instance, if a cut is detected within a dissolve, depending upon the length of the dissolve and the location of the cut within it, it may be decided either to keep only one of them or to keep both but moving one extremity of the dissolve so that it occurs completely before or after the cut.

1.4 Global tuning parameters

The system has several thresholds that have to be tuned for an accurate detection. Depending upon their values, the system can detect or miss more transitions. These thresholds also have to be well balanced among themselves to produce a consistent result. Most of them were manually tuned as the system was built in order to produce the best possible results using development data.

For the TREC-11 and following evaluations, as well as for other applications of the system, we decided to have all the threshold parameters be a function of a global parameter controlling the recall versus precision tradeoff (or, more precisely, the silence to noise ratio). We actually used two such global parameters: one for the cut transitions and one for the gradual transitions. A function was heuristically devised for each system threshold for how it should depend upon the global parameters.

Ten values were selected for the global parameters. These values were selected so that they cover all the useful range (outside of this range, increasing or decreasing further the global parameter produces a loss on both the silence and noise measures) and within that range they set targets on a logarithmic scale for the silence to noise ratio.

1.5 Results

Ten runs have been submitted for the CLIPS-IMAG system. These correspond to the same system with a variation of the global parameter controlling the silence versus noise (or precision versus recall) tradeoff.

Figure 2 shows the relative variation of precision and recall of the SBD system with the global system parameter that controls the silence to noise ratio. Results are shown for all transitions and separately for cuts and gradual transitions. Figure 3 shows the same for frame-precison and frame-recall within detected gradual transitions.

The CLIPS-IMAG system appears to be quite good for gradual transitions both for their detection and location. The F1 measure (harmonic mean of precision and recall) is of 0.727 when the global tuning parameter is set so that precision and recall have comparable values while the best system has an F1 of 0.818. This indicates that the chosen method (comparison of the first and second temporal derivative of the images) is quite good even if theoretically suited only for sequences with no or very little motion. We observe in figure 3 that when the recall inreases and the precision decreases for gradual transition detection, the frame recall remains quite constant while the precision decreases. This may be due to the fact that when

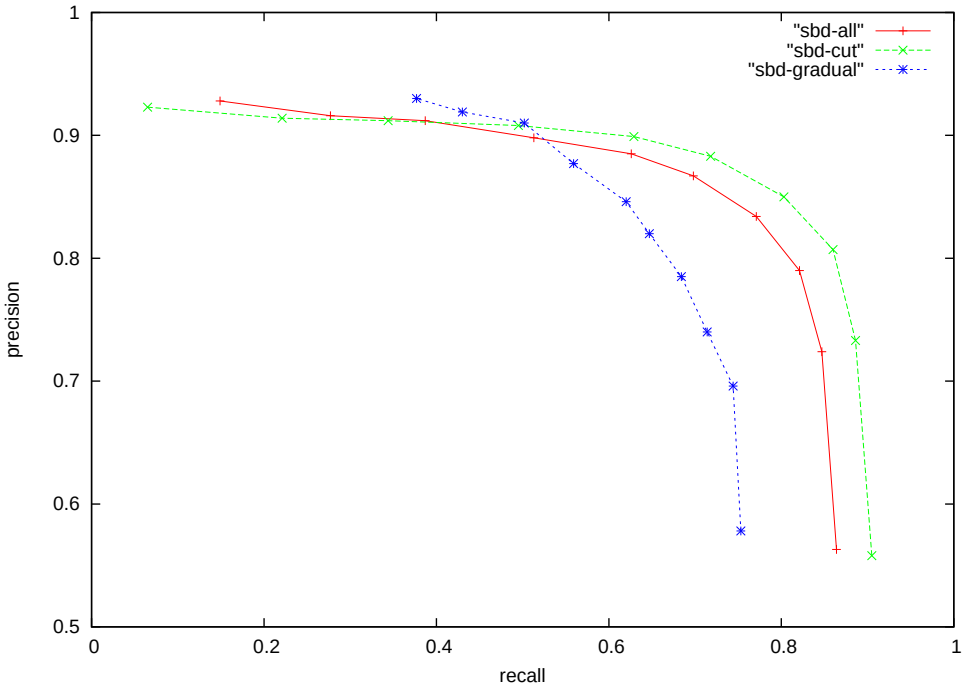

Figure 2: Precision × recall of the CLIPS Shot Boundary Detection system while varying the global system parameter that controls the silence to noise ratio

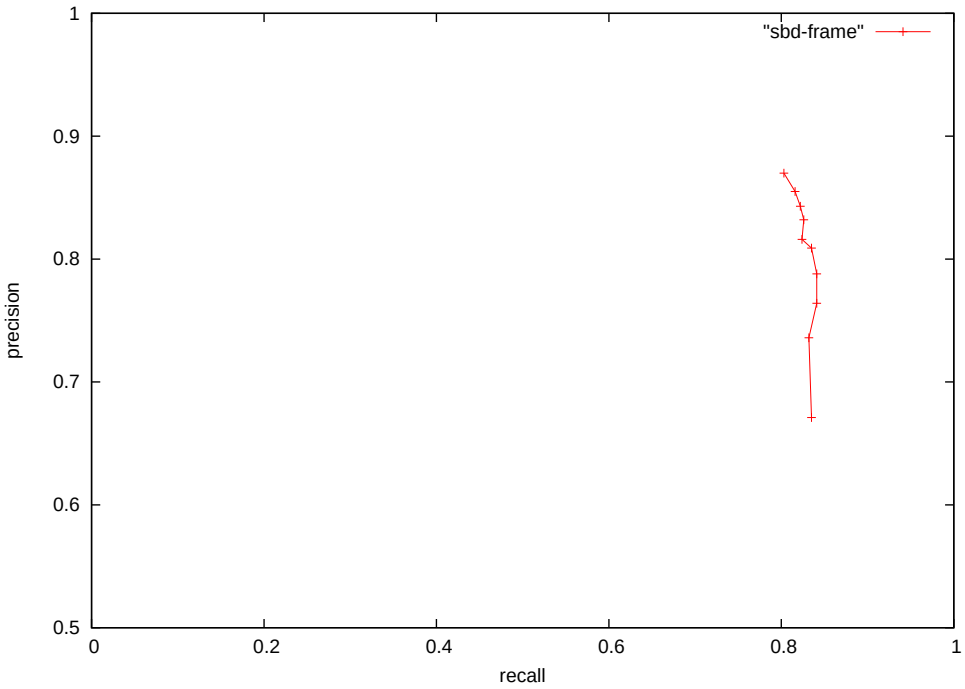

Figure 3: Frame precision × recall for gradual transitions of the CLIPS Shot Boundary Detection system while varying the global system parameter that controls the silence to noise ratio

the recall of gradual transitions is increased, more difficult transitions are considered.

For cuts, the CLIPS-IMAG system has an F1 of 0.833 when the global tuning parameter is set so that precision and recall have comparable values while the best system has an F1 of 0.900. The motion-compensated immage difference coupled to photographic flashes eilimiation is still a good method for shot segementation.

Finally, for all transitions, the CLIPS-IMAG system has an F1 of 0.805 when the global tuning parameter is set so that precision and recall have comparable values while the best system has an F1 of 0.875. The systems is still quite good even though it is a few year old and it has not been optimized on TRECVID 2005 or 2006 data. The golbal control parameter was again very efficient for adjusting the precision versus recall tradeoff.

2 High Level Feature Extraction

High level feature detection was performed using networks of SVM classifiers arranged in a variety of architectures and taking into account a variety of low level descriptors combining text, local and global information as well as conceptual context. The 39 concepts are derived from "intermediate" concepts, themselves derived from "low level" descriptors and not necessarily related to the 39 "final" concepts. This apporach is linked to the idea that it may be better to cross the semantic gap in several steps in which the complexity remains low and the correlation between the inputs and the outputs is kept high. It is related to stacking approaches [2]

2.1 "Local" intermediate concepts

"Local" intermediate concepts are computed on image patches. There are 260 (20×13) approximately half-overlapping 32×32 pixels patches. 15 intermediate concepts were learned each by a single classifier (the same classifier is applied to all the patches within an image) that takes as imputs:

- 9 color components (RGB means, variances and covariances)
- 24 texture components (8 orientations $\times$ 3 scales Gabor transform)
- 7 motion components (the central velovity components plus the mean, variance and covariance of the velocity components within the patch; a velocity vector is computed for every image pixel using an optical flow tool [1] on the whole image).

The intermediate concept classifiers are trained from positives and negative samples extracted from manually labelled image regions.

The 15×260 outputs of the 15 classifiers applied to the 260 patches of the test image are the inputs for the higher level classifiers (the 39 classifiers corresponding to the 39 TRECVID 2006 concepts or intermediates stages of them). In pratcice, not all of the 15 intermediate concepts are used for all of the 39 concepts but only a subset of them. This subset is manually chosen for each of the 39 concepts and typically contains 5 or 6 intermetidate concepts. These 5 to 6×260 components are completed by visual features at the image level (that do not depend upon the intermediate concept; this is actually already an "early" fusion of the patch-concepts and the global low-level image descriptors). These include:

- 64 color components ($4 \times 4 \times 4$ color histogram)
- 40 texture components (8 orientations $\times$ 5 scales Gabor transform)
- 5 motion components (the mean, variance and covariance of the velocity components within the image)

Therefore, the "intermediate" "local" descriptor typically contains about 1500 components. The vector components corresponding to patchs-concepts are not binary but a real value between 0 and 1 corresponding to the estimated probability of the patch of containing the concept as they are computed by the libsvm package [11].

The 15 intermediate concepts considered are: Animal, Building, Car, Cartoon, Crowd, Fire, Flag-US, Greenery, Maps, Road, Sea, Skin_face, Sky, Sports, Studio_background. They have been

learned from the collaborative corpus annotation of TRECVID 2003 and 2005 that we cleaned up and enriched.

2.2 "Reuters" intermediate concepts

"Reuters" intermediate concepts are computed on audio segments of the ASR-MT transcription. They are then projected on the keyframes, each keyframe receiving the values associated to the audio segment in which it is included or the value associated to the nearest ausio segment if it if not included in an audio segment.

The 103 hierarchical Reuters categories have been learned using a Rocchio type classifier using a tf.idf weighting of terms from the 810,000 annotated news samples of the RVC1 Reuters corpus [13]. The classifier is the applied to the speech segments and each one receive a score (also real value between 0 and 1) for each of the 103 reuters category (hiererchy is actually ignored). These 103 values are finally the component of the "intermediate" "reuters" descriptor.

2.3 "Text" intermediate concepts

"Text" intermediate concepts are also computed on audio segments of the ASR-MT transcription. A list of 2500 terms associated to the concepts (these are directly the 39 final ones) is built considering the most frequent ones excluding stopwords. The "intermediate" "text" descriptor is a boolean vector whose components are 0 ore 1 if the term is absent or present in the audio segment. Again the vectors built at the level of the audio segments are projected on the keyframes in the same way.

2.4 Fusion schemes

The descriptors correponding to the different intermediate concepts may be used simultaneously. We tried and compare several fusion schemes for that purpose. This is a difficult problem as the various descriptors differ in nature, in quality and in component count. We consider three fusion schemes inspired from the usual early and late fusion schemes [3] and some variations of them. Those schemes use a classifier to learn the relations between modality components at different abstraction levels.

2.4.1 Early and late fusion

Figure 4 and 5 describe the process of early and late fusion schemes. The feature extraction (FE) process extracts and create a vector for each modality of the video item. We show the SVM process as two main steps: first, the construction of the Kernel, then the Learning or Classification (L / C) processes aims to assign a classification score to the video item.

Merging all the descriptors into a single flat classifier leads to a fully integrated fusion strategy since the fusion classifier obtains all the information from all sources. The advantage of such a scheme is its capacity to learn the regularities formed by the components independently from the modalities. Also, it is easy to use as it just consists in concatenating the various data in a single vector. The main disadvantage is the use of a unique approach (classifier and/or kernel) to merge different types of information. Using an SVM classifier with RBF kernels, an early fusion scheme is equivalent to multiply the kernels which share the same σ parameter. Assuming two concatenated vectors $\mathbf{x}$ and $\mathbf{y}$ from sets of features 1 and 2, we have the following kernel:

$$K(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} = e^{-\frac{\|\mathbf{x}_1-\mathbf{y}_1\|^2+\|\mathbf{x}_2-\mathbf{y}_2\|^2}{2\sigma^2}} = e^{-\frac{\|\mathbf{x}_1-\mathbf{y}_1\|^2}{2\sigma^2}} e^{-\frac{\|\mathbf{x}_2-\mathbf{y}_2\|^2}{2\sigma^2}}$$

The σ parameter is often fixed by using cross validation, it is then optimal for the concatenated vectors, but not necessary for each modality.

A late Fusion is performed on top of several classifiers. It has been presented using different formalisms, such as meta-classification which aims to re-classify the classification results made by other classifiers [4]. The closest theory to illustrate a late Fusion is the Stacking Ensemble learning [2] which is part of the ensemble methods [5]. The idea behind Ensemble learning methods (e.g. bagging, boosting, stacking) is to improve the generalization by training more than one model on each problem (e.g. train 10 SVM instead of just

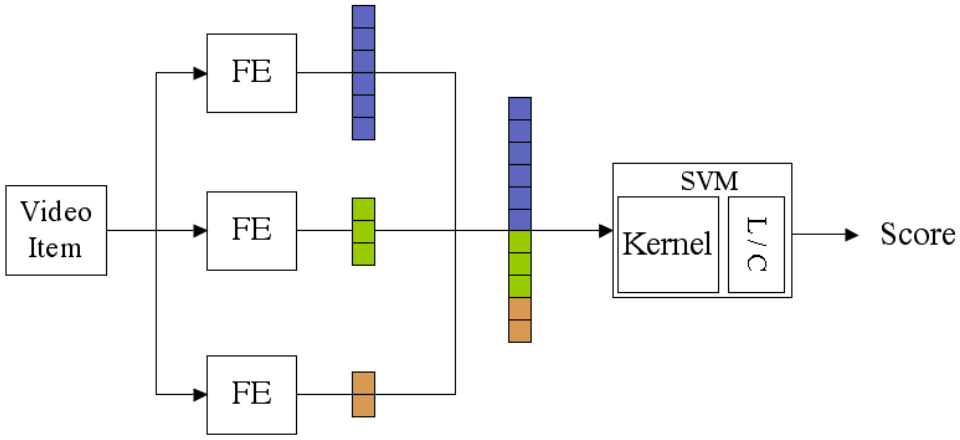

Figure 4: "Early" fusion scheme

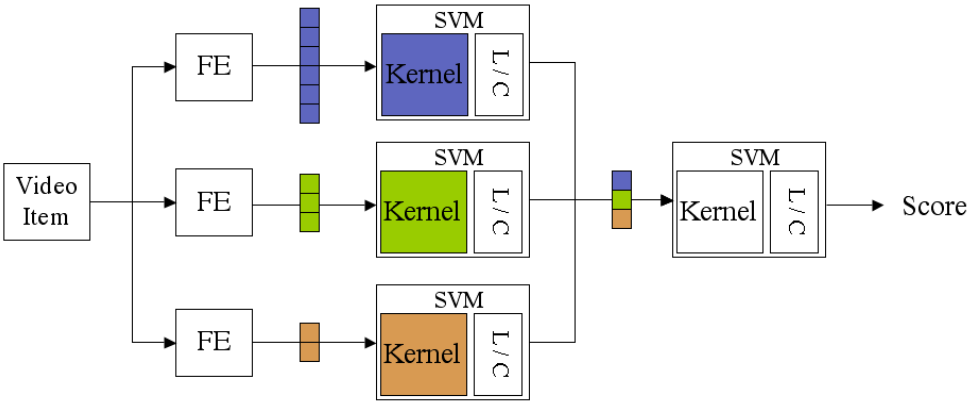

Figure 5: "Late" fusion scheme

one) and then to combine their predictions by averaging, by voting or by other methods. Using stacking, the combination is achieved by a final classifier which provides the final result. Hence, in the context of multimedia indexing, the late fusion scheme consists in performing a first classification separately on each modality and then in merging the outputs using a higher level classifier. In such a way, in contrast with the early fusion, one can use different classifier algorithms and different training sets according to the modalities. Furthermore, the late fusion scheme also allows to combine various classifiers for the same modality. However, the significant dimensional reduction induced by the stacked classifiers might be a disadvantage as the fusion classifier cannot fully benefit from the correlation among the sources of information.

2.4.2 Kernel Fusion

Kernel combination is a current active topic in the field of machine learning. It takes benefit of Kernel-based classifier algorithms. Advantages of merging modalities at kernel level are numerous. First, it allows to choose the kernel functions according to the modalities. For instance, histograms of colors can take advantage of specific histogram matching distances. Likewise, textual modality can be categorized using appropriate kernels such as String Kernels [6] or Word-Sequence kernels [7].

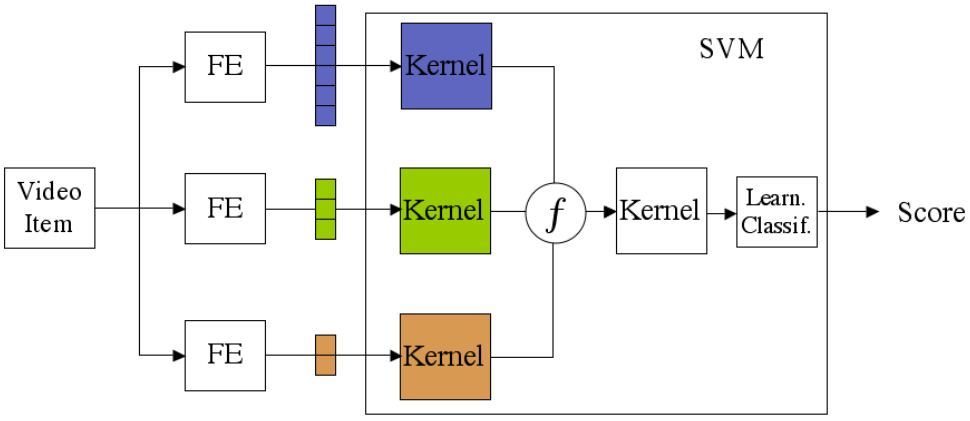

Figure 6: "Kernel" fusion scheme

Kernel fusion also allows to model the data with more appropriate parameters. Merging modalities using an early fusion scheme leads to model the data using a single kernel function. Consequently, when using a RBF kernel, a single σ parameter is expected to "fit" properly the sample vectors relations, whereas it makes much more sense to train a combined RBF kernel using one σ per modality. Combination of unimodal kernels leads to keep as much information as possible from each modality. A combined RBF kernel has the following form:

$$K_c(\mathbf{x}, \mathbf{y}) = F(K_m(\mathbf{x}_m, \mathbf{y}_m)_{(1 \leq m \leq M)})$$

where $K_c(\mathbf{x}, \mathbf{y})$ is the combined kernel value for samples $\mathbf{x}$ and $\mathbf{y}$, $(K_m)_{1 \leq m \leq M}$ are the considered unimodal RBF kernels, F is the combining function over the M modalities, $\mathbf{x}_m$ and $\mathbf{y}_m$ are the sample vectors for modality m. Figure 6 shows the kernel fusion process, the unimodal kernels are merged using a fusion function in order to create the multimodal kernel. Then, learning and classification steps aims to assign a classification score to the video item.

One of the main issues in the current kernel research is the learning of such combined kernels. Called *Multiple Kernels Learning*, it aims to learn at the same time the parameters of all the unimodal kernels and the parameters of the combining function [8]. In our experiments, we used a very simple strategy to create combined kernels. The following algorithm describes the steps to simply create combined kernels:

1. Construct each unimodal kernels K_m,
2. Perform cross-validation on each unimodal kernels to fix their parameters,
3. Construct the combined kernel using the F combining function,
4. Perform cross-validation to optimize the parameters of F.

This algorithm assumes that the best parameters of unimodal kernels are suitable enough to allow efficient generalization of the combined kernel.

Combining individual kernels using a product operator is hightly comparable to the classic early scheme where feature vectors are just concatenated. Assuming two samples $\mathbf{x}$ and $\mathbf{y}$ from sets of features 1 and 2, and using RBF individual kernels, a product combination leads to the following kernel:

$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} = e^{-\frac{\|\mathbf{x}_1-\mathbf{y}_1\|^2+\|\mathbf{x}_2-\mathbf{y}_2\|^2}{2\sigma^2}} = e^{-\frac{\|\mathbf{x}_1-\mathbf{y}_1\|^2}{2\sigma^2}} e^{-\frac{\|\mathbf{x}_2-\mathbf{y}_2\|^2}{2\sigma^2}}$$

The early fusion scheme is equivalent to multiply kernels which share the same σ parameter. Furthermore, due to the product operator, this combination might lead to sparse kernels and provide poor generalization. We used the sum operator instead of the product operator to try to avoid too sparse kernel representations. Summing unimodal

kernels should be more suitable for concept detection when extracted features from a single modality are noisy and lead to incorrect detection.

We actually combine unimodal kernels by linear combination (weighted sum). Using RBF unimodal kernels, combined kernels are defined by the following formula:

$$K_c(\mathbf{x}, \mathbf{y}) = \sum_m w_m \, e^{-\frac{\|\mathbf{x}_m - \mathbf{y}_m\|^2}{2\sigma_m^2}}$$

where σ_m is the RBF parameter of kernel m and w_m is the weight of the associated modality. The w_m's can be fixed *a priori* or by cross-validation. In the conducted experiments, we optimized the w_m's on the training set.

2.5 Normalized Early Fusion

The number of extracted features depends upon the modalities and the type of the features. Hence, an early fusion scheme based on simple vector concatenation is much affected by the vector which have the highest number of inputs. Such fusion should have an impact on the classification, especially with a RBF kernel which is based on Euclidian distance between each training sample.

In traditional SVM implementation, a normalization process is integrated and aims to transform each input in the same range (e.g. [0..1], [−1..1]) in order to unbiased the Euclidian distance. But, for the scope of merging features, this normalization doesn't take into account the number of input from individual features. The goal of normalized early fusion scheme is to avoid the problem of imbalanced features input by reprocessing each feature vectors before concatenation. We normalized each individual vector so that its average norm is about the same. The normalization formula becomes:

$$x_i\prime = \frac{x_i - min_i}{(max_i - min_i) \times \sqrt{Card(x)}}$$

where x_i is an input of the feature vector x, min_i and max_i are respectively the minimum and maximum value of the i^{th} input among the training samples and $Card(x)$ is the number of dimension for the vector x.

2.6 Contextual-Late Fusion

Usual late fusion scheme first classify each concept using individual modalities and then merge the scores in a second layer of classifier. Here, we generalize this scheme by considering more than a single concept. Contextual information has been widely exploited in multimedia indexing [9, 10]. Here, the second layer (stacked) classifier is able to exploit contextual relation between the different concepts. This proposed scheme merges each unimodal classification score from a set of several concepts, in order to exploit both multimodal and conceptual contexts.

Assume that we have M modalities (e.g. visual, audio and text) and C concepts (e.g. Car, Face, Outdoor, Bus, etc). The stacked classifier merges M scores to classify the C concepts in the classic late fusion scheme. The late context fusion scheme merges $M \times C$ classification scores to classify the C concepts.

2.7 Optimized Fusion

The various fusion schemes presented above are characterized by different features, and should have different behaviour among the 39 concepts. The optimized fusion is inspired by the pathfinder framework [9], where an optimization is performed in order to find the best way to infer a concept. Here, we add a selection process to identify the best fusion scheme for each concept, by evaluate the corresponding MAP on the training set.

2.8 Results

Six official runs were submitted since this was the maximum allowed by TRECVID organizers but we actually prepared thirteen of them. The unofficial runs were prepared exactly in the same conditions and before the submission deadline. They are evaluated in the same conditions also using the tools and qrels given by the TRECVID organisers. The only difference is that they did not participate to the pooling process (which is statistically a slight disadavantage).

Table 2.8 gives the inferred average precision (IAP) of all our runs. We experienced different

strategies including the choice of the intermediate features, and the fusion scheme. The official runs are the numbered ones and the number corresponds to the run priority. The IAP of our first run is 0.088 which is slightly above the median while the best system had an IAP of 0.192.

Number	Run	IAP
1	local-reuters-scale	0.0884
2	local-text-scale	0.0864
3	local-reuters-kernel-sum	0.0805
4	local-reuters-kernel-prod	0.0313
5	optimized-fusion-all	0.0674
6	local-reuters-late-context	0.0753
-	local-reuters-early	0.0735
-	local-reuters-late	0.0597
-	local-text-early	0.0806
-	local-text-late	0.0584
-	local	0.0634
-	reuters	0.0080
-	text	0.0106

Table 1: Inferred Average Precision for the high level feature extraction task; "-": not within th official evaluation

2.8.1 Unimodal runs

We observe that the visual and text-based unimodal runs are very different in terms of accuracy; the visual based classification is about 6 times better than the best text based concept detection. This is probably due to the nature of the assessed concepts which seems to be hard to detect using text modality. This point is actually interesting for the evaluation of the ability of the various fusion schemes to handle such heterogeneous data. The features we want to merge lead to different accuracies and are also imbalanced regarding the number of input features.

2.8.2 Classic Early and Late fusion schemes

The two classical fusion schemes do not merge unimodal features similarly. While early fusion is able to outperform both unimodal runs, the late fusion scheme achieves poorer accuracy than the visual run. It might be due to the low number of dimensions handled by the stacked classifier. The early fusion scheme exploits context provided by all of the local visual features and the textual features. The gain obtained by such fusion means that those two modalities provide distinct kind of information. The merged features are, somehow, complementary.

2.8.3 Early based fusion schemes

The gain obtained by the normalized fusion schemes is the most important compare to other fusion schemes. Processing the unimodal features by re-equilibrating them according to the number of dimensions is determinant in order to significantly outperform unimodal runs. In such a way, despite the different number of dimensions, both the visual and textual modalities have the same impact on concept classification. This normalization process leads to a gain of almost 17% (in IAP) comparing to the classic early fusion scheme, which simply normalize input in a common range, and 28% comparing to the better unimodal run.

The gain obtained by the kernel fusion scheme is less significant than the gain obtained by the normalized fusion run. However, when comparing to the classic early fusion, it seems that a combination using sum operator leads to better accuracy than multiplying kernels (which is somehow what the classic early fusion do). Furthermore, it is important to notice that the σ parameters are selected first by cross-validation on unimodal kernels and that we optimize then separately the linear combination. We can expect that an integrated framework which learn simultaneously σ_m and w_m parameters should lead to better results.

2.8.4 Contextual-Late fusion scheme

Contextual-Late fusion is directly comparable with the classical late fusion scheme. This fusion scheme take into account the context from the score of other concepts detected in the same shot. By doing so, the context from other concepts leads to a gain of 26%. Furthermore, we observe that the MIAP obtained using the late contextual fusion scheme is almost the same as the one obtained for the classical early fusion scheme. In

order to go further in this study, it could be interesting to evaluate the impact of the number and/or accuracy rate of concepts used in the context.

We notice that both of unimodal runs lead to poorer accuracy than the median of TRECVID'06 participants. This may be due to the basic and not so optimized features used in our experiments. However, the gain induced by the three fusion schemes presented in this paper lead to better accuracy than the median. We think that an optimization in the choice of descriptors for each modality could enhance the accuracy rate of both unimodal and multimodal runs.

2.8.5 Optimized fusion scheme

Surprisingly, the optimized fusion scheme didn't outperform the other fusion schemes. We identified that this is because of a misstake in our optimization procedure : the optimization has been perform on the training set, which was also used for the learning of the various fusion schemes. Thus, the optimal fusion schemes were found by overfitting the data.

3 Search

The CLIPS-IMAG search system uses a user-controlled combination of five mechanisms: keywords, similarity to example images, semantic categories, similarity to already identified positive image, and temporal closeness to already identified positive image (Figure 7).

The system outputs an ordered list of relevant shots for each topic after interaction with the user (initial query and multiple relevance feedback). The reference segmenation include shots and subshots and was generated by HHI [14]. The system computes a score for each subshot according to the user query and feedback and assign a score to each shot simply as the best score of all its subshots.

3.1 Keyword based search

The keyword based search is done using a vector space model. The words present in the ASR-MT transcription are used as vector space dimensions. Stemming ans stopword list are used. Relevance is first assigned to speech segments (as provided in the ASR-MT transcription) and projected onto overlapping shots.

3.2 Similarity to image examples

Visual similarity between key frames and image examples is looked for using color and texture characteristics. The same primary vector descriptors than for the feature extraction task are used (4×4×4 color histograms and 8×5 Gabor transforms). Distance are computed, normalized and then turned into a relevance value for each characteristic. A 65% color and 35% texture linear combination is then used.

3.3 Feature based search

The goal of this part is to help focusing on specific categories of the video shots, according to a non-crisp labeling of their keyframes. All keyframes are automatically labeled according to the 39 categories used in the high level feature task.

3.4 Visual similarity to already identified positive images

Visual similarity to already retrieved images can be used for the search. These images have to be marked as positive examples for similarity based search by the user (relevance feedback). The search is performed in the same way as for the original image examples. Key frames are ranked according to their closeness to thes positive examples. The images selected for similarity-based search need not to be actually positive example for the current search.

3.5 Temporal closeness to already identified positive images

Temporal closeness (within the video stream) to already retrieved images can be used for the search. These images have to be marked as positive examples for similarity based search by the user (relevance feedback). Key frames are ranked according to their temporal closeness to thes positive examples. The images selected for similarity-based search need not to be actually positive example for the current search.

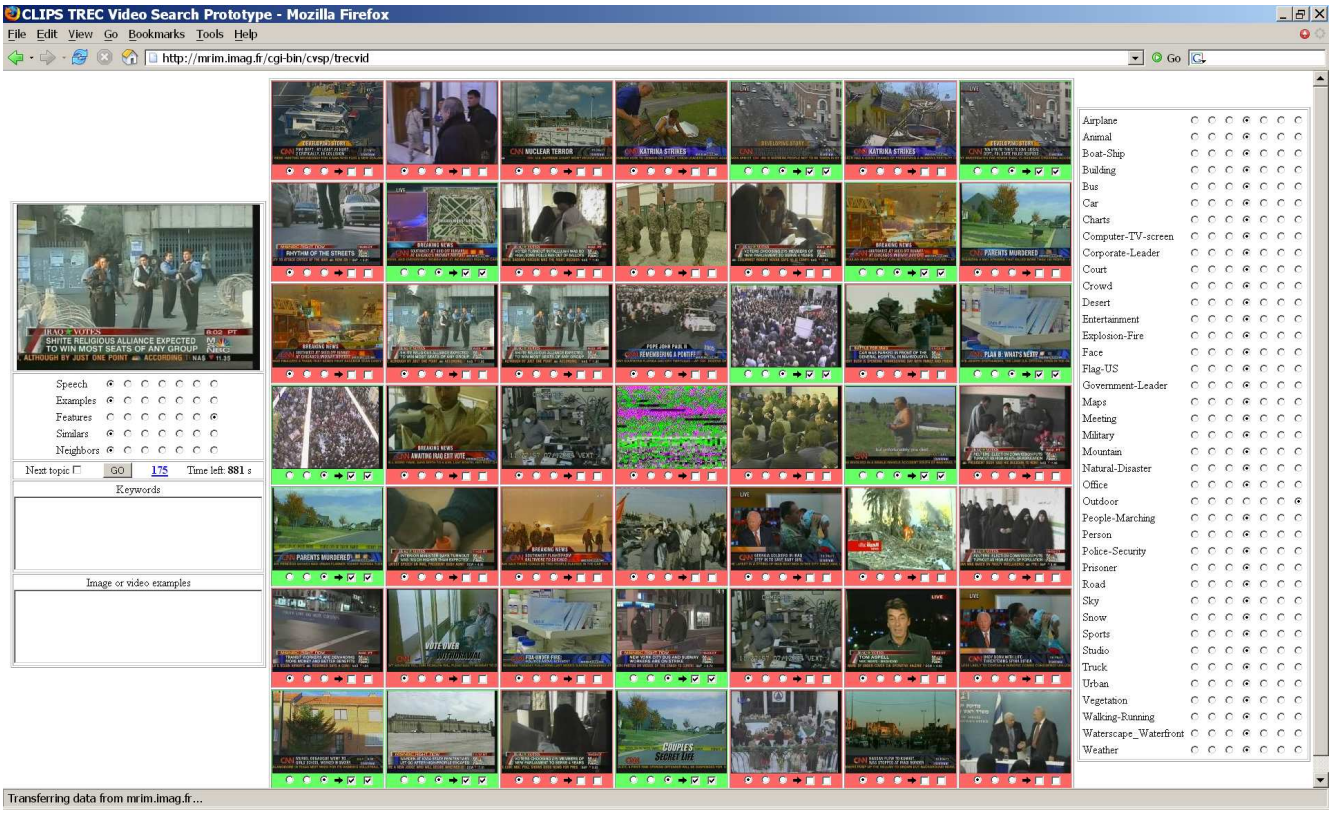

Figure 7: View of the CLIPS-IMAG search system

3.6 Combination of search criteria

The user can define dynamically his search strategy according to the topic and/or the looking of the retrieved images. Each search mechanism can be configured independently and each mechanism can be given a global weight for the search (Figure 7). Relevance are computed independently for each mechanism and for each key frame (or subshot). The per-mechanism relevances are then linearly combined according to the mechanism weight to produce the final key frame relevance. A relevance is computed for each shot at the maximum of the relevances associated to each key frame (or subshot). A ranked list of shots is the produced.

3.7 Search strategy

The system is designed for very fast response time and efficient user feedback. The user is encouraged to use whatever search mechanism seems best appropriate and to view and mark as many images as possible in the given time (900s). At each iteration, the system displays 49 images. By default they are marked as negative. The user only has to mark the positive that he sees by clicking on them. In case of doubt he can see them at actual size in a separate window just by mouse overlap and, if still necessary, he can play the shot by clicking below the images. By default also, the positive images are also positive examples for visual similarity and temporal closeness based search but this can be changed also by the user. Any key frame marked positive by the user receives a relevance of 1 and any key frame marked positive by the user receives a relevance of 0.

The same system has been used for manual and interactive submissions. Manual submissions are the results of the system at the first iteration (without any feedback). Interactive submissions are the results of the system after as many iteration as possible within the allocated time. The system keep track of the output (ranked list of

1000 shots) at each iteration as well as the time elapsed since the beginning of the topic processing. This allows to display the evolution of the Mean Average Precision (MAP) over time during the search.

3.8 Results

Four users have participated to the tests. Some of them did not have the time to process all topics and other (new) users completed the processing of the remaining topics. Each user processed each topic at most once. Table 3.8 shows the Mean Average Precision for each user for manual (a single iteration, no feedback) and interactive searches. For comparison, the best interactive system has a MAP of 0.303 and the median is has a MAP of 0.163.

User	Manual	Interactive
1	0.0354 (5)	0.140 (1)
2	0.0295 (6)	0.184 (2)
3	0.0374 (-)	0.167 (3)
4	0.0261 (-)	0.120 (4)

Table 2: Mean Average Precision for the search task; the MAP value is followed by the TRECVID run number in parentheses; "-": not within th official evaluation

It can be noticed that there is a significant variability of the system performance according to the user. The relative user performance is consistent with the knowledge and the experience the user has of the system. It is also most probable that the mother language as well as the cultural background of the users significantly affect the system/user performance. None of the users here is an English native speaker. None of them either is much familiar with the politics and sports in the US.

Ths Search system is instrumented so that it keeps track of all the system intermediate results each time the user clicks on the "search" button. It is therefore possible, at each time between 0 and 900 seconds to obtain the best results list obtained by the system and the user at that time and it is possible to compute the corresponding system and user performance (Mean Average Precision) at that time. It is then possible to display the performance of the system and of the user as a function of the time. Figure 8 shows the corresponding plots for the four users that participated to the tests.

4 Conclusion

We have presented the systems used by CLIPS-IMAG and LSR-IMAG laboratories for their participation to TRECVID 2006 and the obtained results.

Shot boundary detection was performed using a system based on image difference with motion compensation and direct dissolve detection. This system gives control of the silence to noise ratio over a wide range of values and for an equal value of noise and silence (or recall and precision), the F1 value is 0.805 for all types of transitions, 0.833 for cuts and 0.727 for gradual transitions.

High level feature detection was performed using networks of SVM classifiers arranged in a variety of architectures and taking into account a variety of low level descriptors combining text, local and global information as well as conceptual context. The inferred average precision of our first run is 0.088.

The search system uses a user controlled combination of five mechanisms: keywords, similarity to example images, semantic categories, similarity to already identified positive images, and temporal closeness to already identified positive images. The mean average precision of the system (with the most experienced user) is 0.184.

5 Acknowledgments

This work has been spported by the ISERE CNRS ASIA-STIC project and the Video Indexing INPG BQR project.

References

[1] Quénot, G.M.: Computation of Optical Flow Using Dynamic Programming, In *IAPR Workshop on Machine Vision Applications*, pages 249-52, Tokyo, Japan, 12-14 nov. 1996.

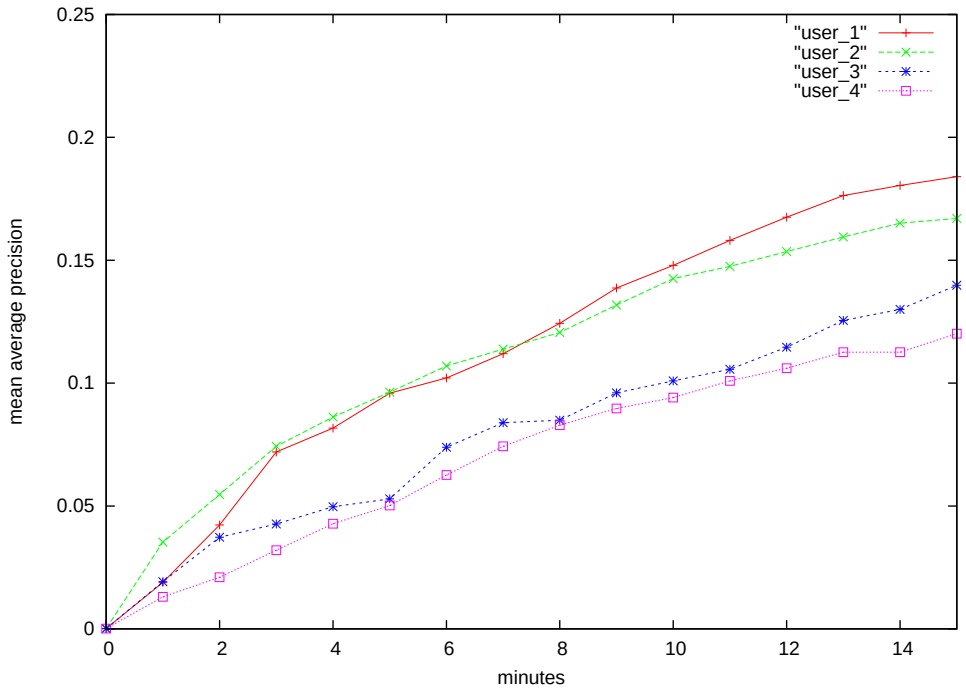

Figure 8: Performance of the CLIPS-IMAG search system and test users over time

[2] D.H. Wolpert Stacked Generalization. Neural Networks, Vol. 5, pp. 241-259, Pergamon Press.

[3] C.G.M. Snoek and M. Worring and A.W.M. Smeulders. Early versus Late Fusion in Semantic Video Analysis. Proceedings of ACM Multimedia, 2005.

[4] W. Lin, R. Jin, and A. Hauptmann. Meta-classification of multimedia classifiers. In *Proceedings of First International Workshop on Knowledge Discovery*.

[5] T. G. Dietterich. Ensemble methods in machine learning. In *Lecture Notes in Computer Science*, 2000.

[6] H. Lodhi, J. Shawe-Taylor, N. Cristianini, and C. J. C. H. Watkins. Text classification using string kernels. *NIPS*, 2000.

[7] N. Cancedda, E. Gaussier, C. Goutte, and J.-M. Renders. Word-sequence kernels. *Journal of Machine Learning Research*, 2003.

[8] S. Sonnenburg, G. Ratsch, and C. Schafer. A general and efficient multiple kernel learning algorithm. *In proceedings of NIPS*, 2005.

[9] C. G. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, F. J. Seinstra, and A. W. Smeulders. The semantic pathfinder for generic news video indexing. In *Proceedings of ICME*, 2006.

[10] M. Naphade. On supervision and statistical learning for semantic multimedia analysis. *Journal of Visual Communication and Image Representation*, 2004.

[11] C. Chang and C Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

[12] S. Ayache, G. Quénot and S. Satoh. Context-based Cconceptual Image Indexing, In ICASSP, 2006.

[13] Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361-397, 2004. http://www.jmlr.org/-papers/volume5/lewis04a/lewis04a.pdf.

[14] C. Petersohn. "Fraunhofer HHI at TRECVID 2004: Shot Boundary Detection System", TREC Video Retrieval Evaluation Online Proceedings, TRECVID, 2004 URL: www-nlpir.nist.gov/-projects/tvpubs/tvpapers04/fraunhofer.pdf