The MediaMill TRECVID 2007 Semantic Video Search Engine

C.G.M. Snoek, I. Everts, J.C. van Gemert, J.M. Geusebroek, B. Huurnink, D.C. Koelma, M. van Liempt,
O. de Rooij, K.E.A. van de Sande, A.W.M. Smeulders, J.R.R. Uijlings, M. Worring
Intelligent Systems Lab Amsterdam, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
http://www.mediamill.nl

Abstract

In this paper we describe our TRECVID 2007 experiments. The MediaMill team participated in two tasks: concept detection and search. For concept detection we extract region-based image features, on grid, keypoint, and segmentation level, which we combine with various supervised learners. In addition, we explore the utility of temporal image features. A late fusion approach of all region-based analysis methods using geometric mean was our most successful run. What is more, using MediaMill Challenge and LSCOM annotations, our visual-only approach generalizes to a set of 572 concept detectors. To handle such a large thesaurus in retrieval, an engine is developed which automatically selects a set of relevant concept detectors based on text matching, ontology querying, and visual concept likelihood. The suggestion engine is evaluated as part of the automatic search task and forms the entry point for our interactive search experiments. For this task we experiment with two browsers for interactive exploration: the well-known CrossBrowser and the novel ForkBrowser. It was found that, while retrieval performance varies substantially per topic, the ForkBrowser is able to produce the same overall results as the CrossBrowser. However, the ForkBrowser obtains top-performance for most topics with less user interaction. Indicating the potential of this browser for interactive search. Similar to previous years our best interactive search runs yield high overall performance, ranking 3rd and 4th.

1 Introduction

Most commercial video search engines such as Google, Blinkx, and YouTube provide access to their repositories based on text, as this is still the easiest way for a user to describe an information need. The indices of these search engines are based on the filename, surrounding text, social tagging, or a transcript. This results in disappointing performance when the visual content is not reflected in the associated text. In addition, when the videos originate from non-English speaking countries, such as China, Lebanon, or the Netherlands, querying the content becomes even harder as automatic speech recognition results are so much poorer. Additional visual analysis yields more robustness. Thus, in video retrieval a recent trend is to learn a lexicon of semantic concepts from multimedia examples and to employ these as entry points in querying the collection.

Last year we presented the *MediaMill 2006* semantic video search engine [28] using a 491 concept lexicon. For our current system we extended it to a thesaurus of 572 concepts. The items vary from pure format like a detected *split screen*, or a style like an *interview*, or an object like a *telephone*, or an event like a *press conference*. Any one of those brings an understanding of the current content. The elements in such a thesaurus offer users a semantic entry to video by allowing them to query on presence or absence of content elements. For a user, however, selecting the right topic from the large thesaurus is difficult. We therefore developed a suggestion engine that analyzes the textual topic given by the user, to automatically derive the most relevant concept detectors for querying the video archive. In addition, we developed novel browsers that present retrieval results using advanced visualizations. Taken together, the *MediaMill 2007* semantic video search engine provides users with semantic access to video archives.

The remainder of the paper is organized as follows. We first define our semantic video indexing architecture in Section 2, emphasizing spatio-temporal visual analysis. Then we highlight our semantic video retrieval engine in Section 3, which includes novel methods for concept suggestion, visual querying, and various video browsers.

2 Semantic Video Indexing

Our generic semantic video indexing architecture is based on the semantic pathfinder [28, 29]. It is founded on the observation that produced video is the result of an authoring process. The semantic pathfinder selects the best path through content analysis, style analysis, and context analysis. This year we again use a semantic pathfinder that relies mainly on (visual) content analysis. In this section we will highlight which components and experiments of last year have been replaced by more elaborate analysis, learning, and combination schemes.

2.1 Supervised Learners

We perceive concept detection in video as a pattern recognition problem. Given pattern $\vec{x}$, part of a shot i, the aim is to obtain a probability measure, which indicates whether semantic concept ω_j is present in shot i. Here shot segmentation is based on [23]. Similar to previous years, we use the Support Vector Machine (SVM) framework [34] for supervised learning of concepts. Here we use the LIBSVM implementation [2] with radial basis function and probabilistic output [22]. We obtain good SVM parameter settings by using an iterative search on a large number of SVM parameter combinations. We optimize SVM parameters that aim to balance positive and negative examples (w_{+1} and w_{-1}). In addition, we also take the γ parameter into account.

We measure performance of all parameter combinations and select the combination that yields the best performance. We use a 3-fold cross validation on the entire 2007 development set to prevent overfitting of parameters. Rather than using regular cross-validation for SVM parameter optimization, we employ *episode-constrained* cross-validation method, as this method is known to yield a more accurate estimate of classifier performance [9].

In addition to the SVM we also experiment with Fisher's linear discriminant [6]. While this classifier is known to be less effective than SVM, in terms of concept detection performance, it requires no parameter tuning so classification is relatively cheap. The Fisher's linear discriminant assumes normal distribution. It is used to find the linear combination of features which best separates two classes. It minimizes the errors in the least square sense. We use the resulting combinations as a linear classifier. For the Fisher classifier we use the PRTools implementation [3]. All classifiers yield a probability measure $p(\omega_j|\vec{x}_i)$, which we use to rank and to combine concept detection results.

2.2 Visual-Only Analysis

Similar to last year's efforts we have concentrated on visual-only analysis. As we observed that regional image features are especially effective for concept detection, our visual analysis emphasizes three types of local image regions: 1) a regular grid; 2) interest points; and 3) segmentation blobs. For TRECVID 2007, we have also conducted a preliminary set of coarse experiments to verify if motion information can be of added value.

On each region level, we aim to decompose complex scenes in proto-concepts like vegetation, water, fire, sky etc. These proto-concepts provide a first step to automatic access to image content [36]. Given a fixed vocabulary of proto-concepts, we assign a similarity score to all proto-concepts for all regions in an image. Different combinations of a similarity histogram of proto-concepts provide a sufficient characterization of a complex scene.

In contrast to codebook approaches [4, 24, 32, 33, 36], we use the similarity to all vocabulary elements [8]. A codebook approach uses the single, best matching vocabulary element to represent an image patch. For example, given a blue area, the codebook approach must choose between water and sky, leaving no room for uncertainty. Following [8], we use the distances to all vocabulary elements. Hence, we model the uncertainty of assigning an image patch to each vocabulary elements. By using similarities to the whole vocabulary, our approach is able to model scenes that consist of elements not present in the codebook vocabulary.

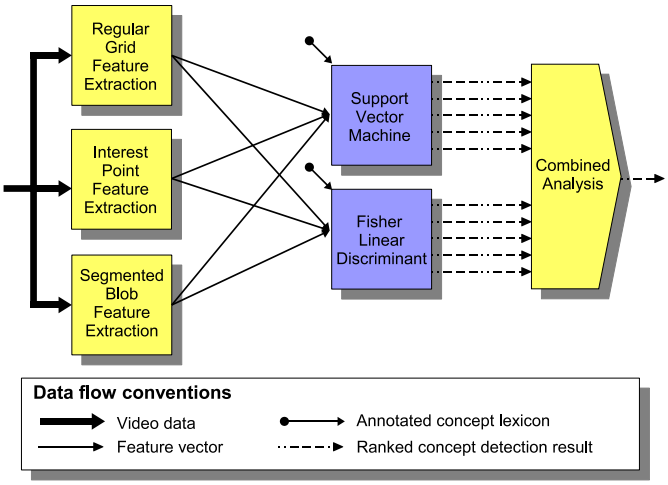

Figure 1: Simplified overview of our visual-only analysis approach for TRECVID 2007. Temporal image feature extraction (not shown) was an independent component.

All visual features are used in isolation or in combination, with the two supervised learners. Finally, we combine the individual concept detectors in several ways and select the combination that maximizes validation set performance. We highlight the major components of our TRECVID 2007 system in Fig. 1.

2.2.1 Image Feature Extraction on Regular Grids

The regular grid is constructed by dividing an image in $n \times n$ overlapping rectangular regions. The overlap between regions is one half of the region size. The number of regions is governed by a parameter r, that indicates the number of regions per dimension, where the two dimensions in the image are the width and height. For example, if $r = 2$ then the overlap between regions leads to 3 regions for both the width and the height, thus yielding $3 \times 3 = 9$ regions.

Wiccest Features We rely on Wiccest features for image feature extraction on regular grids. Wiccest features [11] utilize natural image statistics to effectively model texture information. Texture is described by the distribution of edges in a certain image. Hence, a histogram of a Gaussian derivative filter is used to represent the edge statistics. It was shown in [10] that the complete range of image statistics in natural textures can be well modeled with an inte-

grated Weibull distribution. This distribution is given by

$$f(r) = \frac{\gamma}{2\gamma^{\frac{1}{\gamma}}\beta\Gamma(\frac{1}{\gamma})} \exp\left\{-\frac{1}{\gamma}\left|\frac{r-\mu}{\beta}\right|^{\gamma}\right\}, \quad (1)$$

where r is the edge response to the Gaussian derivative filter and $\Gamma(\cdot)$ is the complete Gamma function, $\Gamma(x) = \int_0^\infty t^{x-1}e^{-1}dt$. The parameter β denotes the width of the distribution, the parameter γ represents the 'peakness' of the distribution, and the parameter μ denotes the mode of the distribution. The position of the mode is influenced by uneven illumination and colored illumination. Hence, to achieve color constancy the values for μ is ignored.

The Wiccest features for an image region consist of the Weibull parameters for the color invariant edges in the region. Thus, the β and γ values for the x-edges and y-edges of the three color channels yields a 12 dimensional descriptor. The similarity between two Wiccest features is given by the accumulated fraction between the respective β and γ parameters: $\sum\left(\frac{\min(\beta_F,\beta_G)}{\max(\beta_F,\beta_G)}\frac{\min(\gamma_F,\gamma_G)}{\max(\gamma_F,\gamma_G)}\right)$, where F and G are Wiccest features. We compute the similarity to 15 proto-concepts [8] for F and G. We divide an input frame into multiple overlapping regions, and compute for each region the similarity to 15 proto-concepts [8]. This yields regional image feature vector **W** for a 2×2 grid and **W2** for a 4×4 grid.

Gabor Features In addition to the Wiccest features, we also rely on Gabor filters for regional image feature extraction. Gabor filters may be used to measure perceptual surface texture in an image [1]. Specifically, Gabor filters respond to regular patterns in a given orientation on a given scale and frequency. A 2D Gabor filter is given by:

$$\widetilde{G}(x,y) = G_\sigma(x,y)\exp\left\{2\pi i\begin{pmatrix}\Omega_{x_0}\\ \Omega_{y_0}\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix}\right\}, \quad i=\sqrt{-1}, \quad (2)$$

where $G_\sigma(x,y)$ is a Gaussian with a scale σ, $\sqrt{\Omega_{x_0}^2+\Omega_{y_0}^2}$ is the radial center frequency and $\tan^{-1}(\frac{\Omega_{y_0}}{\Omega_{x_0}})$ the orientation. Note that a zero-frequency Gabor filter reduces to a Gaussian filter.

In order to obtain an image region descriptor with Gabor filters we follow these three steps: 1) parameterize the Gabor filters 2) incorporate color invariance and 3) construct a histogram. First, the parameters of a Gabor filter consist of orientation, scale and frequency. We use four orientations, $0°, 45°, 90°, 135°$, and two (scale, frequency) pairs: (2.828, 0.720), (1.414, 2.094). Second, color responses are measured by filtering each color channel with a Gabor filter. The $\mathcal{W}$ color invariant is obtained by normalizing each Gabor filtered color channel by the intensity [12]. Finally, a histogram is constructed for each Gabor filtered color channel, where we use histogram intersection as a similarity measure between histograms. We divide an input frame into multiple overlapping regions, and compute for each region the similarity to 15 proto-concepts [8]. This yields regional image feature vector **G** for a 2×2 grid and **G2** for a 4×4 grid.

2.2.2 Image Feature Extraction on Interest Points

Inspired by the work of Zhang [38], we also compute invariant descriptors based on interest points. We use the Difference-of-Gaussians interest point detector by Lowe [19]. The region around the interest point is summarized using one of our SIFT or color descriptors. The SIFT descriptor [19] is consistently among the best performing interest region descriptors [20, 38]. SIFT describes the local shape of the interest region using edge histograms. To make the descriptor invariant, while retaining some positional information, the interest region is divided into a 4x4 grid and every sector has its own edge direction histogram (8 bins). Our color descriptors include color histograms in different color spaces and color extensions of SIFT. For details, we refer to van de Sande [26].

The indexing method used by Zhang involves a comparison between all images, which is not feasible on TRECVID data. Instead, we cluster in descriptor space on descriptors of up to 1,000 positive images of a concept. For all 36 TRECVID concepts we search for at least 10 clusters. Depending on the descriptor and the data clustered on, we obtain between 360 and 400 clusters. We use the improvement over the standard codebook model as introduced in Section 2.2 [8]. However, instead of a similarity function, we use the Euclidean distance between the image descriptors and the clusters. Summing all distances yields a fixed-length feature vector $\vec{F}$ of length n, with n equal to the number of clusters. We term this the mixed keypoint feature vector **S**.

2.2.3 Image Feature Extraction on Segmented Blobs

For each key frame we create segmentation blobs using the algorithm of Felzenszwalb and Huttenlocher [5]. The algorithm uses a graph based segmentation technique which minimizes the within-region color differences. As default settings of the algorithm we used a minimum region size of 100 pixels, a Gaussian smoothing parameter σ of 0.8 to remove digitization artifacts, and a threshold k of 100, which influences the number of regions per image. These settings result in approximately 150-300 regions per images where the sizes of individual regions varies greatly. For each segmented blob we compute the Wiccest features and Gabor features, as detailed in Section 2.2.1, yielding **blobW,blobG,** and **blobWG**.

2.2.4 Exploring Temporal Image Feature Extraction

We extract motion information from a video sequence by first detecting interest points in every frame with the Harris-Laplace corner detector, describing them with SIFT [19], tracking these points over time based on feature similarity and representing the obtained tracks. We assume that the

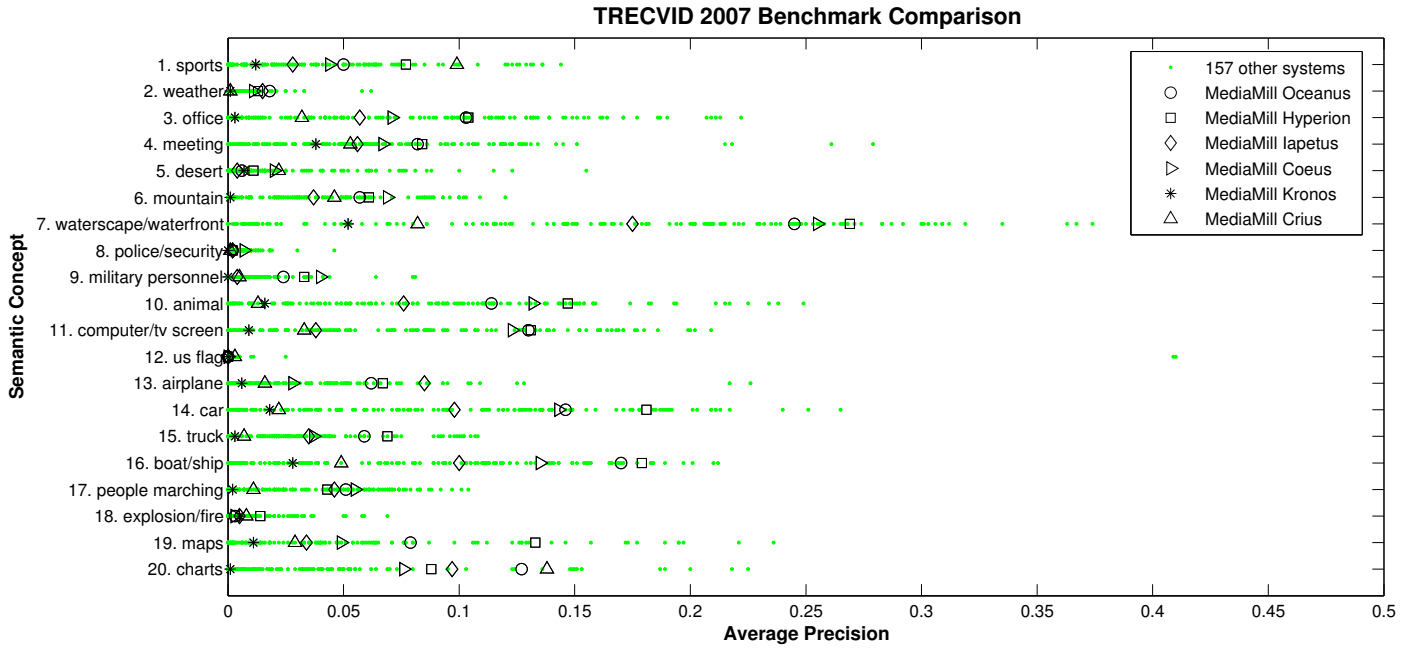

Figure 2: Comparison of MediaMill video indexing experiments with present-day indexing systems in the TRECVID 2007 benchmark.

interest point detector will detect the same points in subsequent frames when differences between them are small. Points can appear and vanish. We address this problem by considering feature tracking from frame to frame to be an assignment problem. To optimize the assignment we use the Hungarian algorithm [17, 35]. Note that we do not impose any constraint on the possible movements of points, and also do not perform any prediction of point locations based on a motion model, as we want our features to be generic.

Having processed the whole video sequence, T tracks are obtained consisting of series of point locations. We represent these tracks by quantizing the displacement from frame to frame in 2-dimensional motion histograms, constructed such that the center bin represents no or very little motion. We paste the R rows of the histogram into one (RxR)x1 feature vector to obtain the final motion feature per track.

In our experiments we follow the approach of clustering the raw features [16] (thereby creating 'motion codebooks', inspired on [18]), projecting the features per shot on the clusters to obtain a single feature vector, learning concepts from the projected features and evaluating with cross validation on the development set. We have conducted experiments using only motion features, only SIFT features, and one in which we fused the two feature types (before learning). Cross validation results (data not shown) indicate that a combination of motion and sift features performs marginally better than using them individually.

2.3 Submitted Concept Detection Results

An overview of our submitted concept detection results is depicted in Fig. 2. We relied on the annotation provided by the 2007 MCG-ICT-CAS team. We will detail each submitted run below.

2.3.1 Run 'Kronos': Temporal Image Features

Although the issues relating track length, time coverage and maximal speed led to intuitively imposed constraints, they have severely effected our experiments as all the tracks from about half the number of shots were rejected and therefore we have processed only half of the data. Apart from too strict constraints, this is also due to a non perfect shot segmentation. Note that the performance of the SIFT features alone is not as good as one might think due to the data sparsity. All these issues make it very difficult to interpret the results. For now, we take our cross validation results (data not shown) as a clue, that a combination of visual and motion features can indeed improve performance, baring in mind the coarseness and simplicity of this experiment. In coming experiments we will address the problems that have arisen, and introduce more features based on motion that are useful for generic content based video indexing.

2.3.2 Runs 'Iapetus' and 'Crius': Cross-Domain Generalization

With the current increase of digital video, it is advantageous to have a content-based indexing system that generalizes well over heterogeneous sets of video collections. One factor that influences generalization is the supervised classifier that requires positive and negative annotations. These annotations are time consuming to obtain, and it is not guaranteed that annotations of the same concept generalize over video domains. To test our generalization performance

over video domains, we compare two sets of annotations, while keeping the type of features constant. To constrain the experiment, we focus on visual analysis only. The best performing feature, based on cross-validation performance on this years data, is an early fusion of **W2** and **G2**, yielding the **WG2** feature. Hence, we submitted two runs based on the **WG2** features. One run (Iapetus) is trained on the data and annotations of this year, where the other run (Crius) is trained on the data and annotations of the MediaMill Challenge [31] using TRECVID 2005 data.

The results show that in most cases, detectors trained on the 2007 development set outperform those trained on 2005 data. However, when the concept detectors trained on 2005 data outperform those trained on 2007 they often obtain our highest overall performance. For detectors of concepts like *charts, us flag, desert* and *sports* the number of available learning examples in the MediaMill Challenge outperform those provided for TRECVID 2007, which could explain the difference in performance. However, for the concept *airplane* this explanation does not hold, as only 29 examples are available in 2007, but still it outperforms the 2005 detector which uses 428 examples (data not shown). We are currently exploring how a combination of annotations over multiple domains influences concept detector performance.

2.3.3 Run 'Coeus': Mixed Keypoint Features

This run uses interest point feature extraction combined with different descriptors. To make interest point description more robust, we also include the overlapping regions approach as an 'interest region' detector. With two detectors (Difference-of-Gaussians and overlapping regions) and six region descriptors, we have a total of twelve different experiments for our mixed keypoint method. For each of the twelve combinations of region detectors and descriptors we have applied both Fisher and SVM classifiers, yielding 24 ranked lists of shots. For late fusion of such ranked lists several methods exist, e.g., min, max, sum, median, and product [7]. An extension of product fusion that is capable to handle missing data is the geometric mean. We found after several experiments on MediaMill Challenge data that this geometric mean outperforms the other fusion methods. Hence, we combine the various lists using the geometric mean. For a single shot i the combined likelihood becomes:

$$\exp\left[\frac{1}{n}\sum_{k=1}^{n}\ln p_k(\omega_j|\vec{x}_i)\right], \qquad (3)$$

where n equal to the number of experiments, in our case up to five experiments, selected on a per-concept basis using cross validation. The advantage of the geometric mean is its ability to handle a variable number of likelihoods per shot. If the n varies between shots, the geometric means of those shots can be compared. We use this property for shots which do not have any interest regions: these shots have no likelihood, but if at least one combination has a likelihood for this shot, then we are able to compute a geometric mean. For the concepts *mountain*, *police security*, *military* and *people marching*, this run is our top contender.

2.3.4 Run 'Oceanus': Fusion of Visual Experiments through Feature Selection

This run is a late fusion of our experiments based on visual features. For the 36 TRECVID concepts all our experiments mentioned before, excluding temporal image features, are candidates for inclusion. Last year however, fusing *all* experiments did not yield good results on MediaMill Challenge data. Instead, we choose to use cross-validation performance to the best one-third experiments per concept. The fusion of the different experiments is again performed using the geometric mean from eq. (3). The fusion of visual-only analysis results is our second-best overall run. For the *weather*, the feature selection method performs better than all our other runs. Feature selection gives similar performance as the selection of all experiments for *office*, *meeting* and *computer/TV screen*. For *water* and *animal*, mixed interest points end up behind the 'Hyperion' run which fuses all experiments, while the feature selection is outperformed. This suggests that the mixed interest point experiments were under-represented in the experiments selected in the 'Oceanus' run. Therefore, we believe that our feature selection strategy leaves room for further improvement.

2.3.5 Run 'Hyperion': Fusion of All Visual Experiments

This run is a late fusion of all our experiments, excluding temporal image features. For the 36 TRECVID concepts all experiments are included. The fusion of the different experiments is again performed using the geometric mean from eq. (3). The fusion of all visual-only experiments is our best overall run. For the concepts *waterscape/waterfront*, *animal*, *car*, *truck*, *boat/ship*, *explosion* and *maps* the 'Hyperion' run outperforms our other runs. For the concepts *office*, *meeting* and *computer/TV screen* our fusion runs achieve near-identical performance. For the majority of the concepts, our combination runs perform best, suggesting that single visual experiments are not powerful enough. Also, it is interesting that fusion of all experiments (run 'Hyperion') outperforms feature selection (run 'Oceanus'). This suggest that our feature selection strategy can be improved upon.

2.4 Scaling-up to 572 Concept Detectors

We adopt a graceful degradation approach to further scale our lexicon of concept detectors. We use annotations from both the MediaMill Challenge [31] and LSCOM [21], which are provided for TRECVID 2005 video data. And also additional annotations for a *black and white* and *colored* footage. We employ a variation of visual features including several **W**,**G**, and **S** variations in combination with Fisher's linear discriminant. Because parameter optimization of the SVM is expensive, performing a complete analysis for all

concepts was not feasible. We combined individual classification results using geometric mean to yield a single combined ranked result. While the performance might not be optimal, the detectors may still be useful for semantic video retrieval.

3 Semantic Video Retrieval

Our TRECVID 2007 search task efforts have concentrated on automatic and interactive retrieval using the lexicon of 572 learned concept detectors. For users, remembering a list of 572 concepts is not feasible. We therefore developed a query suggestion engine which finds the most appropriate combination of concept detectors and multilingual text retrieval results given the topic. This yields a ranking of the data. A convenient way of browsing the result is our CrossBrowser [30], which allows to use both the rank and temporal context of a shot. There are, however, many other relevant directions which can be explored e.g. different semantic threads through the data or shots visually similar to the current shot. This year we therefore developed the novel ForkBrowser which allows a user to browse multiple directions, while maintaining overview.

3.1 Automatic Search

As concept lexicon size increases, topics are more likely to be strongly associated with multiple concept detectors. Where previously in the automatic search task we concentrated on finding the one best detector for a topic, this year we combined the results from multiple detectors per topic. In our approach we converted detectors from ranked lists to 'binary' detectors, and used these to re-rank a sorted result list. As well as the detectors from the 2007 high level feature task, we experimented with the incorporation of detectors developed on the 2005 development set. We found that these detectors helped boost retrieval performance, despite having been trained on English, Arabic, and Chinese broadcast news rather than Dutch television programs.

The primary ingredients of our automatic search system are:

Dutch speech recognition transcripts Dutch automatic speech recognition(ASR) text provided by the University of Twente [13], with commonly occurring stop words removed and reduced to morphological components using the Dutch version of the Porter stemmer.

English machine translation transcripts English machine translation (MT) text obtained by automatically translating the Dutch speech recognition text, provided by Christof Monz of Queen Mary University of London.

38 TRECVID 2007 detectors 38 detectors trained on TRECVID 2007 development data. This consists of the 36 detectors defined in the 2007 high-level feature task, as well as a black-and-white and a color detector.

363 TRECVID 2005 detectors 363 detectors trained on TRECVID 2005 development data. This consists of detectors defined by LSCOM and MediaMill, with duplicates and rarely occurring concepts removed [27].

In the following subsections we describe our transcript (text) based search, our detector based search, the automatic search runs, and our results.

3.1.1 Retrieval Using Transcripts

In transcript-based search we leveraged both the original Dutch ASR transcripts, as well as the English MT transcripts. Both transcripts were indexed separately. At retrieval time, each topic statement was automatically translated into Dutch using the online translation tool freetranslation.com, allowing a search on the MT transcripts with the original (English) topic text, and a search on ASR transcripts using the translated Dutch topic text. The resulting two ranked lists were then combined to form a single list of transcript-based search results.

To compensate for the temporal mismatch between the audio and the visual channels, we used our temporal redundancy approach [14]. To summarise this approach, the transcript of each shot is expanded with the transcripts from temporally adjacent shots, where the words of the transcripts are weighted according to their distance from the central shot. The adjusted word counts are calculated according to:

$$count(word, shot') = \sum_{n \in N} (\gamma(shot_n) \cdot count(word, shot_n)), \tag{4}$$

where n is the absolute shot offset from the current shot, N is the number of neighbouring shots to be included, and $\gamma(shot_n)$ is the offset-dependent weighting of the $shot_n$. The weight γ is given by $0.9664n^{-1.2763}$, which we previously found to work well for topic retrieval in the TRECVID 2003-2006 data sets. The maximum window size N was set to 15, as performance gains beyond this window size were negligible for the same data sets. These experiments are more fully described in [14].

After the expanded transcripts were indexed, retrieval was done using the language modeling approach with Jelinek-Mercer smoothing [37], a smoothing method that we previously found to be suited to speech-based video retrieval [15].

For retrieval, Dutch speech recognition transcripts and English machine translation transcripts were indexed separately. At retrieval time, the text of each topic was automatically translated into Dutch using the online translation service `freetranslation.com`. The original English topic text was used to search the MT transcript, and the translated Dutch topic was used to search the original ASR transcript. The two lists of results were then combined.

Table 1: Automatic search run MAP scores, with highest MAP highlighted per topic

Topic ID	Topic Summary	UvA-MM3	UvA-MM4	UvA-MM5	UvA-MM6
0197	people walking up stairs	0.0003	0.0007	**0.0119**	0
0198	door being opened	**0.002**	**0.002**	0.0006	**0.002**
0199	person walking or riding a bicycle	0.0148	0.0176	**0.1453**	0.012
0200	hands at a keyboard	**0.0002**	**0.0002**	0	**0.0002**
0201	canal, river, or stream	0.0062	0.0061	**0.0067**	0.006
0202	person talking on a telephone	**0.0008**	**0.0008**	0.0001	**0.0008**
0203	street market scene	**0.0003**	**0.0003**	0	**0.0003**
0204	street protest or parade	0.0094	**0.0098**	0	0.0094
0205	train in motion	**0.1638**	0.1593	0.0116	**0.1638**
0206	hills or mountains	0.0013	0.0022	**0.1064**	0.0013
0207	waterfront with water and buildings	0.014	0.0144	**0.0998**	0.0109
0208	street at night	0.0002	0.0002	**0.010**	0.0002
0209	people sitting at a table	0.0069	0.0090	**0.0265**	0.0069
0210	people walking with one or more dogs	0	0	**0.0001**	0
0211	sheep or goats	0.0226	**0.0246**	0.0041	0.0226
0212	a boat moving past	0.0026	0.0027	**0.0886**	0.0026
0213	woman talking toward the camera in an interview	**0.0047**	0.0045	0.0015	**0.0047**
0214	very large crowd of people	0.0105	0.0113	**0.1827**	0.0001
0215	classroom scene	0.0004	0.0004	**0.0006**	0.0004
0216	bridge	0.0014	**0.0021**	0.0012	0.0012
0217	shots of a road through the front windshield	0.0013	**0.0018**	0.006	0.0016
0218	people playing musical instruments	**0.0448**	0.0447	0.0011	0.0447
0219	Cook character in the Klokhuis series	0.0913	**0.0961**	0	0.0913
0220	grayscale ... street with buildings and people	0.0019	**0.0554**	0.0319	0.0015
MAP		*0.0167*	*0.0194*	***0.0307***	*0.0160*

3.1.2 Retrieval Using Detectors

In detector based retrieval, we were especially concerned with combining results from multiple detectors into a single ranked list. First we used text matching techniques to find potentially useful concepts for answering the topic. Subsequently we used a boosting approach to combine the detectors, starting with a single list of results and then reordering that list using detector results. We describe the concept selection and the detector combination steps in more detail below.

Concept Selection The first step in our detector based search was the identification of concept detectors that might be relevant for a given topic. In order to achieve this we first indexed textual metadata about each concept - the concept name, the concept description given to the annotators, and using the WordNet links obtained in [27] the concept synonyms and glosses. At retrieval time, the topic text was matched against the text in the different concept fields. In addition, hyponym and hypernym concepts were retrieved. Concepts were then ranked according to the extent to which the topic description matched the textual metadata in the various fields.

Detector Combination When combining the detectors, we started with a ranked list of 'trusted results', which could be the results from a detector or from a text search. We then re-ranked this list using the associated concept detectors identified in the concept selection phase. Some detectors are more likely to be helpful for a topic than others, so we incorporated a number of measures to assign a weight to each detector. In the weight assignation we distinguished between a *concept* and the *detector* designed to find that concept. While a concept may be very semantically related to a given topic, the associated detector could be of low quality and therefore not very helpful in finding relevant shots. This also needs to be taken into account. The following measures were used, each being normalised to a number from 0 to 1:

1. Topic-concept match. The extent of the match between the topic text and the concept detection, as described in the Detector Selection section.

2. Detector performance. The detector MAP on the hold-out/validation set, adjusted to take into account the performance of a 'random' detector. For detectors trained on 2005 data, we add an extra penalty factor as we expect performance to degrade on the new data set

3. Concept specificity. A measure to take into account how frequently a concept occurs in the collection, performing a similar function to the 'document frequency' measure in text retrieval. Calculated as $1 - \frac{\text{number of relevant shots}}{\text{total number of shots}}$ using the annotations on the development collection.

For detector combination, we first convert the scored detector lists into binary lists, where each shot is given a

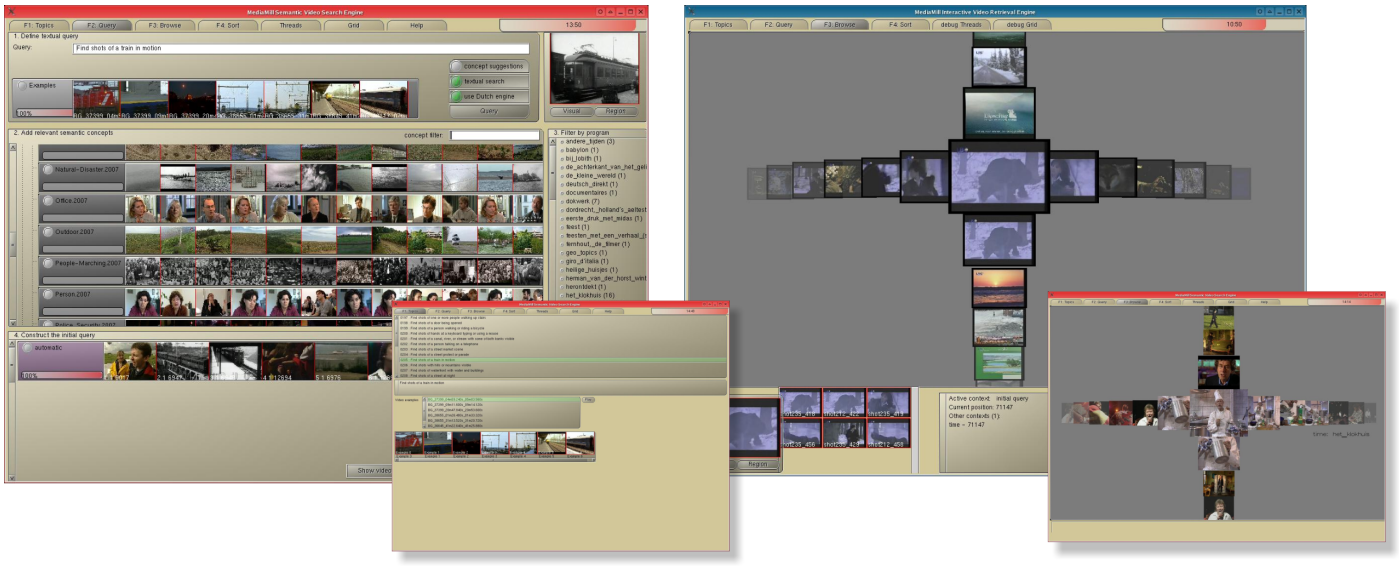

Figure 3: Screenshots of the MediaMill Semantic Video Search Engine with the CrossBrowser.

relevance score of 1 (expected to be relevant), or 0 (expected to not be relevant). The conversion is done by assigning a relevance of 1 to the top $2n$ results of each detector, where $n = \frac{\text{number of relevant shots in training set}}{\text{total number of shots in training set}} \times$ number of relevant shots in current collection. For each shot we then combine the three measures to assign a detector weight, and boost the relevant shots on the list.

3.1.3 Submitted Automatic Search Results

We submitted four runs for automatic search, namely:

UvA-MM6 Text baseline. Retrieval using transcripts, as described in Section 3.1.1.

UvA-MM3 Text + TRECVID 2007 assigned concepts (trained on 2007FSD data) + black and white and colour detectors. Use text as the 'trusted result' list, and boost with detectors.

UvA-MM4 Text + TRECVID 2007 assigned concepts + black and white and colour detectors + MediaMill and LSCOM concepts (trained on 2005FSD data). Use text as the 'trusted result' list, and boost with detectors.

UvA-MM5 Visual baseline. Retrieval using the TRECVID 2007 assigned concepts + black and white and colour detectors + MediaMill and LSCOM concepts. Use best scoring detector as the 'trusted result' list, and boost with remaining detectors.

As can be seen in Table 1, the best performing run was the visual baseline. The worst performing run was the text baseline, which in general yielded very low MAP precision scores. The poor performance of the text baseline is likely due in part to the lack of topics requesting named people and objects, which are more likely to be mentioned in transcripts than more general statements. Another factor may be the decreased quality of the speech recognition transcripts with respect to previous years, as the sound quality of the recordings is often noticeably lower than in the broadcast news data that was used before. It is not apparent from the TRECVID runs whether our combination approach was particularly effective, and we plan to investigate this further in the near future.

3.2 Interactive Search: Video Browsing

In traditional video retrieval systems users may query video archives by keyword, by example, by concept, by time or by program, subsequently they browses through the results, and when the results are unsatisfactory the process reiterates. As a consequence of this iterative process a lot of time is spent on query specification. Moreover, when the target search results are not returned by the system in the initial queries a user may run out of query ideas. To alleviate both problems we try to depart from this traditional approach. We do so by providing users with browsers that allow to visualize the entire data set in multiple dimensions. This facilitates interactive exploration. For TRECVID 2007, we have focussed specifically on consolidation of proven effective interface components from previous TRECVID editions into a novel browsing environment. A graphical overview of our 2007 system is depicted in Fig. 3.

3.2.1 Video Threads

We introduce the notion of threads in order to browse through a video data set in multiple directions. A thread is a linked sequence of shots in a specified order, based upon an aspect of their content [25]. We define two types of threads: *static threads* which are pre-computed beforehand, and *dynamic threads* which are generated on demand during a browse session. The content of a thread is based on a

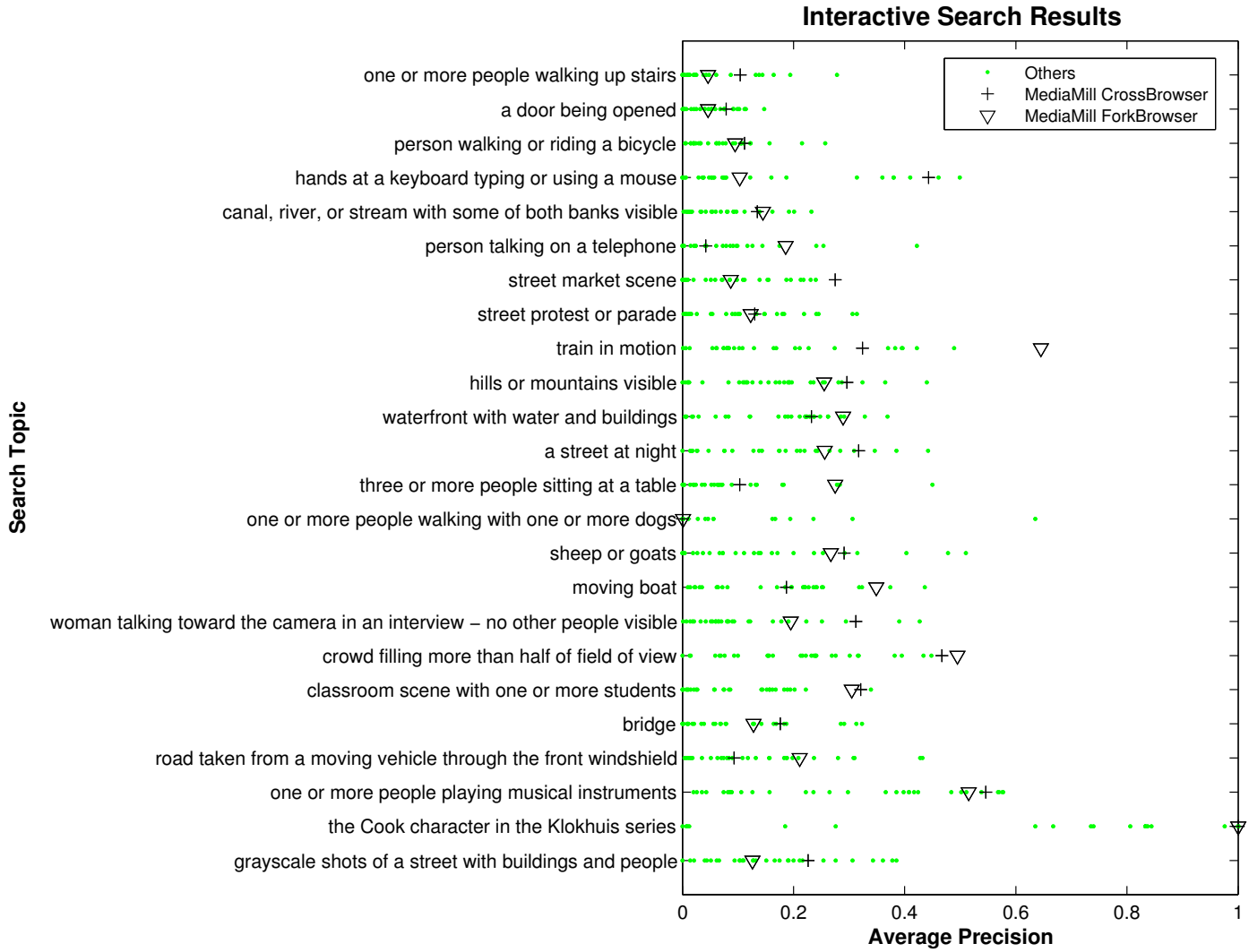

Figure 4: Comparison of interactive video search results for 24 topics performed by 33 users of present-day video retrieval systems. MediaMill results are indicated with special markers.

form of similarity between shots in the data set. The MediaMill 2007 video search engine offers the following threads and similarities.

- *Visual threads*: based on similarity between visual features,
- *Time threads*: based on temporal similarity between shots,
- *Query result threads*: based on similarity between shots and a user posed query,
- *History threads*: based on shots a user has visited;

Each method yields a ranking of the data through which the user has to browse.

3.2.2 Visualizing Threads using the ForkBrowser

Combination of the time thread with any other thread resulted in the CrossBrowser. This browser proved effective for interactive search in 2004 and 2005 when a single thread, for example a single concept detector query, is sufficient to solve the topic [28,30]. For topics that require a combination of threads we introduced the RotorBrowser in 2006 [25,28]. This browser allows to integrate query results with time, visual similarity, semantic similarity and various other shot based similarity metrics. While effective, this visualization proved overwhelming for non-expert users. To leverage the benefits of having multiple query methods while simultaneously allowing the user to maintain overview, we introduce a new interface which combines query by keyword, by example, by 572 concepts, by time and by program into a rigid framework: the ForkBrowser.

The ForkBrowser visualizes results by displaying animated key frames based on the shape of a fork. The contents of the tines of the fork depend on the shot at the top of the stem. The center tine shows unseen query results, the leftmost and rightmost tines show the time thread, and the

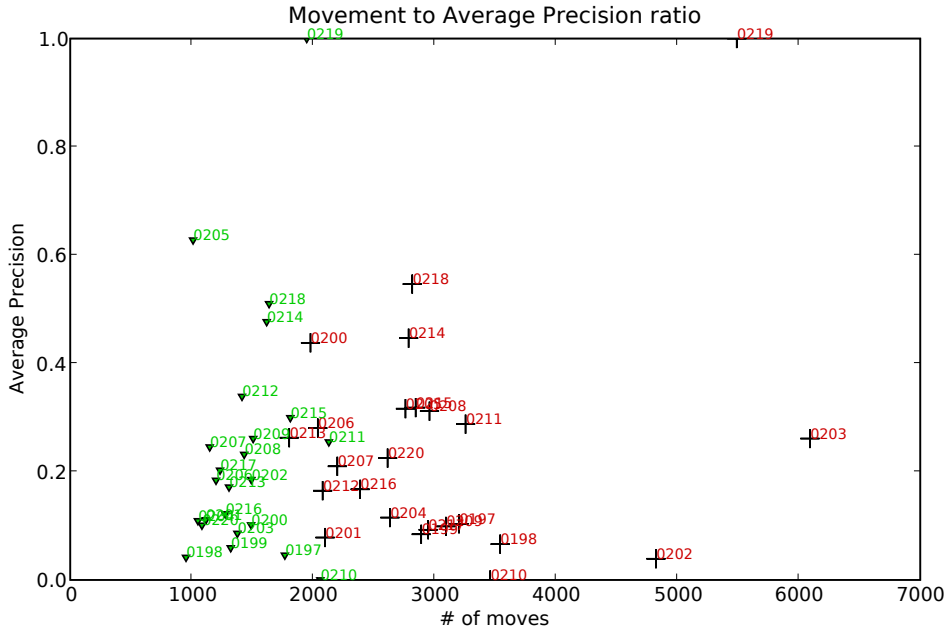

Figure 5: Average Precision versus number of move interactions for both the CrossBrowser (+) and the ForkBrowser (▽).

two tines in between show user assignable threads. For the TRECVID 2007 benchmark we chose to display 2 variants of visual similarity threads here. The stem of the fork displays the history thread. All browse directions, each tine and the stem, are both accessible by keyboard and mouse for quick navigation. Every displayed key frame can also be played on demand by displaying up to 16 frames from the originating shot. This helps answering queries containing explicit motion rapidly.

3.2.3 Submitted Interactive Search Results

We submitted two runs for interactive search, comparing two expert users. One user performed the interactive search by using the MediaMill search engine with the CrossBrowser (**UvA-MM1**). Another user exploited the MediaMill system in combination with the ForkBrowser (**UvA-MM2**).

During the runs the system logged all user actions, which were automatically parsed to provide browse statistics. These include the time users spent adjusting a query, time spent while searching for results, the number of user interaction steps made during searching, the number of results selected, the thread from which results were selected, and so on. Results in Fig. 4 indicate that for most search topics, users of the MediaMill system score above average. Furthermore, users of our approach obtain a top-3 average precision result for 8 out of 24 topics. Best performance is obtained for 4 topics. Both browsers achieve nearly the same mean average precision (CrossBrowser: 0.259, ForkBrowser: 0.256), but average precision scores for individual topics vary greatly. If we compare the number of interaction steps required per topic with the average precision achieved for each topic (see Fig. 5) we observe that the ForkBrowser required significantly less user interaction within the 15 minute time frame. A thorough analysis of the browsing results is underway. A first analysis of ForkBrowser results indicates that each topic required a different combination of threads in order to find good results. See Fig. 6 for an overview of thread usage for a selected set of topics.

Acknowledgments

This research is sponsored by the European VIDI-Video project, the BSIK MultimediaN project, and the NWO MuNCH project. The authors are grateful to NIST and the TRECVID coordinators for the benchmark organization effort.

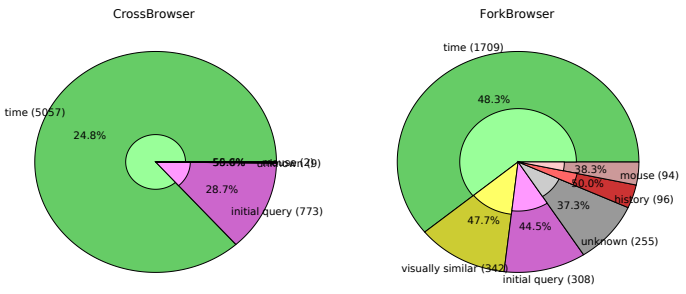

Figure 6: These graphs shows how many results were obtained from each thread, and which percentage of them was judged relevant. Left: CrossBrowser results. Right: ForkBrowser results.

References

[1] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.

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

[3] R. Duin et al. PRTools version 4.0: A matlab toolbox for pattern recognition, 2006. `http://www.prtools.org/`.

[4] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[5] P. Felzenszwalb and D. Huttenlocher. Efficient Graph-Based Image Segmentation. *Int'l J. Computer Vision*, 59(2):167–181, 2004.

[6] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[7] E. Fox and J. Shaw. Combination of multiple searches. In *TREC-2*, pages 243–252, 1994.

[8] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, C. G. M. Snoek, and A. W. M. Smeulders. Robust scene categorization by learning image statistics in context. In *Int'l Workshop SLAM, CVPR'06*, New York, USA, 2006.

[9] J. C. van Gemert, C. G. M. Snoek, C. Veenman, and A. W. M. Smeulders. The influence of cross-validation on video classification performance. In *Proc. ACM Int'l Conf. Multimedia*, pages 695–698, Santa Barbara, USA, 2006.

[10] J.-M. Geusebroek and A. W. M. Smeulders. A six-stimulus theory for stochastic texture. *Int'l J. Computer Vision*, 62(1/2):7–16, 2005.

[11] J.-M. Geusebroek. Compact object descriptors from local colour invariant histograms. In *British Machine Vision Conference*, Edinburgh, UK, 2006.

[12] M. A. Hoang, J. M. Geusebroek, and A. W. M. Smeulders. Color texture measurement and segmentation. *Signal Processing*, 85(2):265–275, 2005.

[13] M. Huijbregts, R. Ordelman, and F. de Jong. Annotation of heterogeneous multimedia content using automatic speech recognition. In *Proc. Int'l Conf. Semantics And digital Media Technologies*, LNCS, Berlin, 2007. Springer Verlag.

[14] B. Huurnink and M. de Rijke. Exploiting redundancy in cross-channel video retrieval. In *Proc. ACM Int'l Workshop MIR*, pages 177–186, Augsburg, Germany, 2007.

[15] B. Huurnink and M. de Rijke. The value of stories for speech-based video search. In *Proc. ACM CIVR*, pages 266–271. Amsterdam, The Netherlands, 2007.

[16] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Int'l Conf. Computer Vision*, 2005.

[17] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[18] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int'l J. Computer Vision*, 43(1):29–44, 2001.

[19] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. Computer Vision*, 60(2):91–110, 2004.

[20] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[21] M. Naphade, J. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 13(3):86–91, 2006.

[22] J. Platt. Probabilities for SV machines. In A. Smola, *et al.*, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

[23] C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot boundary detection system. In *Proc. TRECVID Workshop*, Gaithersburg, USA, 2004.

[24] P. Quelhas, F. Monay, J. M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. V. Gool. Modeling scenes with local descriptors and latent aspects. In *IEEE Int'l Conf. Computer Vision*, 2005.

[25] O. de Rooij, C. G. M. Snoek, and M. Worring. Query on demand video browsing. In *Proc. ACM Int'l Conf. Multimedia*, pages 811–814, Augsburg, Germany, 2007.

[26] K. Sande. Coloring concept detection in video using interest regions. Master's thesis, University of Amsterdam, 2007.

[27] C. G. M. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring. Adding semantics to detectors for video retrieval. *IEEE Trans. Multimedia*, 9(5):975–986, 2007.

[28] C. G. M. Snoek, J. C. van Gemert, T. Gevers, B. Huurnink, D. C. Koelma, M. van Liempt, O. de Rooij, K. E. A. van de Sande, F. J. Seinstra, A. W. M. Smeulders, A. H. C. Thean, C. J. Veenman, and M. Worring. The MediaMill TRECVID 2006 semantic video search engine. In *Proc. TRECVID Workshop*, Gaithersburg, USA, 2006.

[29] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, F. J. Seinstra, and A. W. M. Smeulders. The semantic pathfinder: Using an authoring metaphor for generic multimedia indexing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(10):1678–1689, 2006.

[30] C. G. M. Snoek, M. Worring, D. C. Koelma, and A. W. M. Smeulders. A learned lexicon-driven paradigm for interactive video retrieval. *IEEE Trans. Multimedia*, 9(2):280–292, 2007.

[31] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proc. ACM Int'l Conf. Multimedia*, pages 421–430, Santa Barbara, USA, 2006.

[32] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. In Y. Weiss, *et al.*, editors, *Advances in NIPS*, pages 1299–1306. MIT Press, Cambridge, MA, 2006.

[33] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang. Image classification for content-based indexing. *IEEE Trans. Image Processing*, 10(1):117–130, 2001.

[34] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, USA, 2nd edition, 2000.

[35] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001.

[36] J. Vogel and B. Schiele. Natural scene retrieval based on a semantic modeling step. In *CIVR*, Dublin, Ireland, 2004.

[37] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Information Systems*, 22(2):179–214, 2004.

[38] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int'l J. Computer Vision*, 73(2):213–238, 2007.