The MediaMill TRECVID 2006 Semantic Video Search Engine

C.G.M. Snoek, J.C. van Gemert, Th. 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, 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 2006 experiments. The MediaMill team participated in two tasks: concept detection and search. For concept detection we use the MediaMill Challenge as experimental platform. The MediaMill Challenge divides the generic video indexing problem into a visual-only, textual-only, early fusion, late fusion, and combined analysis experiment. We provide a baseline implementation for each experiment together with baseline results, which we made available to the TRECVID community. The Challenge package was downloaded more than 80 times and we anticipate that it has been used by several teams for their 2006 submission. Our Challenge experiments focus specifically on visual-only analysis of video (run id: B_MM). We extract image features, on global, regional, and keypoint level, which we combine with various supervised learners. A late fusion approach of visual-only analysis methods using geometric mean was our most successful run. With this run we conquer the Challenge baseline by more than 50%. Our concept detection experiments have resulted in the best score for three concepts: i.e. desert*,* flag us*, and* charts*. What is more, using LSCOM annotations, our visual-only approach generalizes well to a set of 491 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 and ontology querying. The suggestion engine is evaluated as part of the automatic search task (run id: A-MM) and forms the entry point for our interactive search experiments (run id: A-MM). Here we experiment with query by object matching and two browsers for interactive exploration: the CrossBrowser and the novel RotorBrowser. It was found that the RotorBrowser is able to produce the same results as the CrossBrowser, but with less user interaction. Similar to previous years our best interactive search runs yield top performance, ranking 2nd and 6th overall. Again a lot has been learned during this year's TRECVID campaign, we highlight the most important lessons at the end of this paper.*

*The Netherlands Organisation for Applied Scientific Research (TNO), Signal Processing Department, Stieltjesweg 1, Postbus 155, 2600 AD Delft, The Netherlands.

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 2005* semantic video search engine [34] using a 101 concept lexicon. For our current system we made a jump to a thesaurus of 491 concepts. The items vary from pure format like a detected *split screen*, or a style like an *interview*, or an object like a *horse*, or an event like an *airplane take off*. 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 2006* semantic video search engine provides users with semantic access to news video archives.

The remainder of the paper is organized as follows. We first define our semantic video indexing architecture in Section 2, introducing the MediaMil Challenge and our mostly visual analysis approach for this year's TRECVID. Then we highlight our semantic video retrieval engine in Section 3, which includes novel methods for concept suggestion, visual querying, and various video browsers. We wrap up in Section 4 where we highlight the most important lessons learned.

2 Semantic Video Indexing

Our generic semantic video indexing architecture is based on the semantic pathfinder [34, 35]. 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 use a semantic pathfinder that relies mainly on (visual) content analysis, where the MediaMill Challenge [37] replaces the content analysis step. In this section we will highlight which components and experiments of the Challenge have been replaced by more elaborate analysis, learning, and combination schemes.

2.1 MediaMill Challenge

TRECVID has been of pivotal importance in assessing complete video indexing methods on their relative merit. In the course of the TRECVID benchmark some groups have shared annotations, like LSCOM [23], donated features, like the camera shot segmentation by CLIPS-IMAG [27], speech recognition results donated by LIMSI [10] and various multimedia features donated by Informedia [46]. In addition, all participants share their results on common test data for a limited lexicon of typically 10 high-level concepts. Until recently, however, nobody has provided low-level features and detected semantic concepts for a large lexicon on both training and test data, while these are crucial assets for repeatability of intermediate analysis steps.

This is mainly caused by the fact that TRECVID focuses on the final result of concept detection systems. In theory, the TRECVID experiments are repeatable, but not on a system component level. Because TRECVID ignores intermediate results, component-based optimization and comparison during methodology development are impossible in practice. To gain insight in intermediate steps that affect performance of concept detection methods, while simultaneously pushing performance to the modular max, we have proposed and distributed the MediaMill Challenge during the 2006 TRECVID benchmark [37].

The Challenge divides the generic video indexing problem into a visual-only, textual-only, early fusion, late fusion, and combined analysis experiment, see Fig.1. We provide a baseline implementation for each experiment together with baseline results for a lexicon containing 101 semantic concept detectors. The 85 hours of training data from the TRECVID 2005 corpus forms the basis for the MediaMill Challenge. We divided this archive a priori into a non-overlapping train and test set. The Challenge train set $\mathcal{A}$ contains 70% of the data, and the Challenge test set $\mathcal{B}$ holds the remaining 30%. The Challenge package has been downloaded more than 80 times, and we anticipate that it has been used by several teams for their 2006 system, either for comparison or as a building block for their submission.

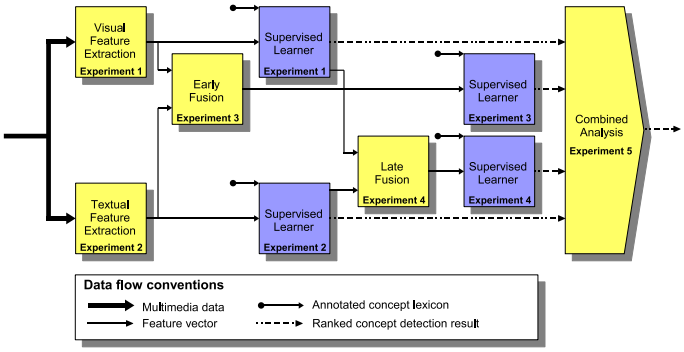

Figure 1: Data flow within the MediaMill Challenge for generic video indexing of 101 semantic concepts [37]. Experiment 1 and 2 focus on unimodal analysis, yielding a visual and a textual concept classification. Experiment 3 and 4 employ an early and late fusion scheme respectively. The Challenge allows for the construction of four classifiers for each concept. In experiment 5, an optimum is selected based on combined analysis.

2.2 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. Similar to the MediaMill Challenge, we use the Support Vector Machine (SVM) framework [44] for supervised learning of concepts. Here we use the LIBSVM implementation [4] with radial basis function and probabilistic output [24]. We obtain good SVM parameter settings by using an iterative search on a large number of SVM parameter combinations. The MediaMill Challenge optimizes SVM parameters that aim to balance positive and negative examples (w_{+1} and w_{-1}). Here we take the γ parameter into account also. We measure average precision performance of all parameter combinations and select the combination that yields the best performance. We use a 3-fold cross validation on Challenge train set $\mathcal{A}$ to prevent overfitting of parameters. Rather than using regular cross-validation for SVM parameter optimization, we also experiment with the recently proposed *episode-constrained* cross-validation method, as this method is known to yield a more accurate estimate of classifier performance [12].

In addition to the SVM we also experiment with logistic regression and Fisher's linear discriminant [8]. While both classifiers are known to be less effective than SVM, in terms of concept detection performance, they require no parameter tuning so classification is relatively cheap. Logistic regression performs a maximum likelihood estimation of weights for the different feature dimensions, under the assumption that the observed training data was generated by a binomial model. In contrast, 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 both classifiers we use the PRTools implementation [5]. All

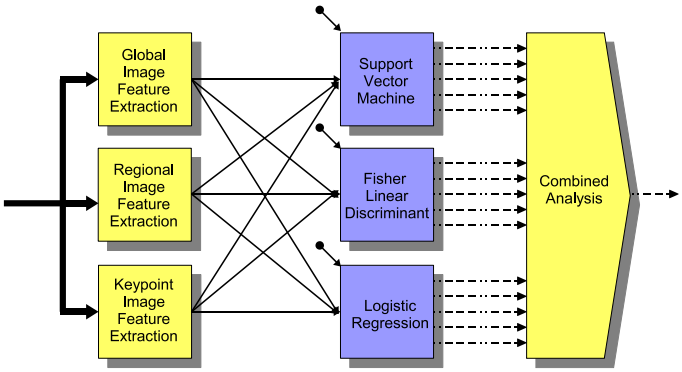

Figure 2: Simplified overview of our visual-only analysis approach for TRECVID 2006, using the conventions of Fig.1.

three classifiers yield a probability measure $p(\omega_j|\vec{x}_i)$, which we use to rank and to combine concept detection results.

2.3 Visual-Only Analysis

Given the promising performance of our visual features in last years benchmark, we have concentrated this years' efforts mainly on visual-only analysis, i.e. experiment 1 of the MediaMill Challenge. We extract image features on three levels of abstraction, namely: global level, region level, and keypoint level. On each 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 [45]. 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 [6, 26, 38, 40, 45], we use the similarity to all vocabulary elements [11]. 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 [11], 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.

All visual features are used in isolation or in combination, with the three supervised learners. Finally, we combine the individual concept detectors in several ways and select the combination that maximizes validation set performance.

2.3.1 Global Image Feature Extraction

We rely on Wiccest features for global image feature extraction. Wiccest features [14] 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. Since there are more non-edge pixels then there are edge pixels, a histogram of edge responses for natural images typically has a peak around zero, i.e.: many pixels have no edge responses. Additionally, the shape of the tails of the distribution is often in-between a power-law and a Gaussian distribution. The tail emphasizes the long-range correlation between edge pixels in the image. A heavy power-law tail indicates a strongly contrasting object-background edge, whereas a Gaussian tail indicates a noisy, high-frequency texture region. The complete range of image statistics in natural textures can be well modeled with an integrated Weibull distribution [13]. This distribution is given by

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

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 integrated Weibull distribution can be estimated from a histogram of filter responses with a maximum likelihood estimator (MLE) as described in [14]. The parameters μ, β and γ are estimated by taking the derivatives of the integrated Weibull distribution to the respective parameters and setting them to zero. The parameters β and γ are dependant on each other, therefore a dichotomic search scheme is utilized to estimate the best β and γ combination. Note that a histogram is only used to speed up the calculations. Therefore, the number of bins in the histogram should be taken as large as the computation time allows.

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 [11] for F and G. This yields global image feature vector $\mathbf{W1}$.

2.3.2 Regional Image Feature Extraction

We also use Wiccest features for regional image feature extraction. We divide an input frame into multiple overlapping regions, and compute for each region the similarity to 15 proto-concepts [11]. This yields regional image feature vector $\mathbf{W}$.

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 [3]. Specifically, Gabor filters respond to regular patterns in a given orientation on a given scale and frequency. A 2D Gabor filter is given by:

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

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 [15]. 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 [11]. This yields regional image feature vector **G**.

2.3.3 Keypoint Image Feature Extraction

Inspired by the work of Zhang [47], we also compute invariant descriptors based on interest regions. In an evaluation of interest region detectors, Mikolajczyk et al [22] found that the Harris-Affine detector performs best. However, Zhang obtains best results using the Harris-Laplace interest region detector, noting that affine invariance can often be unstable in the presence of large affine or perspective distortions.

The Harris-Laplace interest region detector [20] uses a Harris corner detector on an image at multiple smoothing scales to detect keypoints. We compute the Laplacian at scales near the scale at which the keypoint was detected. The scale at which the Laplacian is at a local maximum is selected as the scale of the keypoint. The point is rejected if there is no local maximum of the Laplacian. Detected scale and keypoint together form a circular interest region, which can be detected under rotation and scale changes.

The SIFT descriptor [19] is consistently among the best performing interest region descriptors [21, 47]. 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). The grid is aligned with the dominant direction of the edges in the interest region to make the descriptor rotation invariant.

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 39 TRECVID concepts we search for at least 10 clusters. Depending on the descriptor and the data clustered on, we obtain between 400 and 425 clusters.

We use the improvement over the standard codebook model as introduced in Section 2.3 [11]. 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 keypoint image feature vector **S**.

2.4 Visual-Only Challenge Results

We performed several experiments against the MediaMill Challenge using the feature vectors **W1**, **W**, **G**, and **S** in combination with SVM, logistic regression and Fisher's linear discriminant. In addition to using the global, regional, and keypoint features separately, we also explored their combined influence on concept detection performance using vector concatenation. An overview of the results for the 39 TRECVID concepts is given in Fig. 3.

The Challenge baseline is the SVM with feature vector **W**, yielding a mean average precision (MAP) of 0.250 on the 39 TRECVID concepts (Fig. 3, column 2). Our best overall results are obtained with an SVM and **WG** combination using episode constrained cross-validation and inclusion of the γ parameter. Improving upon the Challenge by 41%. Combining features with an SVM yields better performance than using logistic regression or Fisher's linear discriminant. However, these two classifiers allow for quick classification of relatively long feature vectors. Sometimes even outperforming the best SVM detector for a concept. The Fisher linear discriminant is especially effective in classification tasks that involve long feature vectors. When we select the feature and classifier combination that yields the best performance per concept we may obtain an increase over the Challenge of as much as 48% for the 39 TRECVID concepts. For the complete lexicon of 101 concepts from the Challenge the increase is more than 68% (data not shown).

2.5 Submitted Concept Detection Results

All our experiments were performed on the MediaMill Challenge, including parameter optimization and best-of selection. Since the Challenge is based on TRECVID 2005 training data only, we extended the annotations for our final submission with more positive examples from the TRECVID 2005 test set. These were obtained by manual inspection of last years result. We added the positive feature vectors at model construction time, they were not used for parameter optimization. An overview of our submitted concept detection results is depicted in Fig. 4. We will now highlight the details of each submitted run.

Concept	Support Vector Machine					Logistic Regression								Fisher Linear Discriminant								Best
	W	G	WG	WG+	WG+g	W	G	S	W1	W1W	WG	W1WG	WGS	W1WGS	W	G	S	W1	W1W	WG	W1WG	WGS	W1WGS	
Sports	0.304	0.268	0.370	0.388	**0.434**	0.212	0.212	0.130	0.055	0.273	0.325	0.351	0.329	0.359	0.226	0.219	0.128	0.050	0.266	0.337	0.365	0.364	0.387	0.434
Entertainment	0.166	0.175	0.175	0.329	**0.441**	0.281	0.236	0.262	0.267	0.297	0.310	0.323	0.364	0.371	0.278	0.235	0.261	0.267	0.294	0.300	0.313	0.349	0.355	0.441
Weather	0.405	0.525	0.557	0.556	**0.659**	0.344	0.485	0.210	0.167	0.350	0.547	0.548	0.368	0.307	0.313	0.452	0.235	0.124	0.347	0.524	0.535	0.567	0.577	0.659
Court	0.093	0.133	**0.205**	0.134	0.167	0.071	0.073	0.019	0.021	0.120	0.115	0.107	0.102	0.098	0.048	0.096	0.037	0.010	0.056	0.146	0.151	0.098	0.102	0.205
Office	0.077	0.040	0.070	**0.093**	0.091	0.076	0.047	0.051	0.071	0.092	0.080	0.093	0.086	0.087	0.070	0.045	0.050	0.068	0.082	0.077	0.087	0.077	0.081	0.093
Meeting	0.257	0.259	0.316	0.315	0.309	0.245	0.235	0.173	0.133	0.232	0.310	0.303	0.321	0.309	0.229	0.210	0.167	0.120	0.222	0.313	0.309	**0.335**	0.331	0.335
Studio	0.636	0.698	0.714	0.716	**0.781**	0.640	0.670	0.482	0.520	0.673	0.744	0.754	0.772	0.778	0.618	0.657	0.497	0.478	0.651	0.730	0.744	0.770	0.778	0.781
Outdoor	0.688	0.736	0.758	0.762	0.782	0.739	0.750	0.660	0.674	0.747	0.773	0.779	0.785	0.788	0.740	0.748	0.660	0.675	0.748	0.773	0.779	0.789	**0.793**	0.793
Building	0.316	0.325	0.356	0.364	0.317	0.303	0.312	0.296	0.272	0.312	0.360	0.359	0.375	0.372	0.294	0.324	0.298	0.263	0.307	0.369	0.369	**0.402**	0.402	0.402
Desert	0.103	0.113	0.145	0.155	0.099	0.106	0.101	0.052	0.077	0.103	0.103	0.107	0.046	0.019	0.147	0.109	0.086	0.091	0.158	0.178	**0.198**	0.154	0.164	0.198
Vegetation	0.183	0.303	0.324	0.312	**0.329**	0.215	0.326	0.119	0.147	0.210	0.326	0.323	0.285	0.280	0.199	0.284	0.120	0.154	0.202	0.308	0.312	0.318	0.320	0.329
Mountain	0.141	0.120	0.215	0.203	**0.265**	0.192	0.140	0.092	0.074	0.217	0.262	0.250	0.189	0.186	0.107	0.127	0.091	0.076	0.119	0.163	0.168	0.204	0.209	0.265
Road	0.195	0.236	0.259	0.259	**0.262**	0.190	0.219	0.175	0.168	0.197	0.237	0.239	0.232	0.236	0.186	0.207	0.172	0.162	0.193	0.228	0.229	0.235	0.237	0.262
Sky	0.478	0.499	0.545	0.557	0.577	0.535	0.521	0.401	0.339	0.540	0.588	0.587	0.592	0.590	0.511	0.517	0.403	0.331	0.516	0.587	0.582	**0.609**	0.607	0.609
Snow	0.085	0.118	0.131	0.135	**0.177**	0.054	0.113	0.060	0.019	0.067	0.103	0.112	0.089	0.094	0.044	0.107	0.063	0.018	0.035	0.117	0.102	0.162	0.149	0.177
Urban	0.222	0.234	0.256	0.256	**0.286**	0.217	0.225	0.213	0.192	0.228	0.259	0.263	0.271	0.273	0.212	0.221	0.215	0.193	0.224	0.252	0.256	0.275	0.277	0.286
Waterscape	0.150	0.220	0.241	0.244	**0.399**	0.174	0.184	0.100	0.107	0.177	0.240	0.235	0.219	0.215	0.141	0.173	0.100	0.110	0.148	0.195	0.199	0.215	0.215	0.399
Crowd	0.480	0.502	0.500	0.509	0.546	0.519	0.519	0.440	0.439	0.516	0.559	0.556	**0.588**	0.586	0.494	0.503	0.447	0.429	0.498	0.534	0.536	0.577	0.578	0.588
Face	0.895	0.878	0.909	0.910	0.925	0.897	0.887	0.859	0.839	0.899	0.918	0.919	0.925	**0.925**	0.897	0.886	0.857	0.838	0.899	0.918	0.919	0.925	0.925	0.925
Person	0.831	0.848	0.846	0.946	**0.956**	0.941	0.928	0.915	0.909	0.943	0.950	0.951	0.953	0.953	0.941	0.928	0.913	0.907	0.942	0.949	0.950	0.952	0.952	0.956
Government_leader	0.213	0.218	0.241	0.240	0.230	0.235	0.212	0.223	0.159	0.235	0.260	0.266	0.314	**0.318**	0.234	0.208	0.219	0.158	0.231	0.255	0.258	0.301	0.306	0.318
Corporate_leader	0.016	**0.025**	0.024	0.023	0.017	0.018	0.020	0.021	0.017	0.018	0.021	0.020	0.024	0.023	0.018	0.019	0.020	0.016	0.019	0.020	0.020	0.024	0.024	0.025
Police_security	0.012	0.011	0.020	0.015	**0.189**	0.018	0.012	0.012	0.011	0.017	0.029	0.022	0.023	0.022	0.014	0.011	0.011	0.010	0.012	0.024	0.014	0.018	0.016	0.189
Military	0.217	0.210	0.234	0.237	0.208	0.242	0.242	0.156	0.200	0.258	0.257	**0.270**	0.258	0.265	0.226	0.233	0.147	0.188	0.238	0.249	0.258	0.247	0.256	0.270
Prisoner	0.047	0.182	0.198	0.200	**0.237**	0.005	0.046	0.006	0.006	0.010	0.015	0.021	0.011	0.011	0.006	0.029	0.016	0.004	0.023	0.055	0.085	0.028	0.032	0.237
Animal	0.209	0.217	0.317	0.323	**0.491**	0.149	0.101	0.127	0.054	0.211	0.219	0.255	0.162	0.175	0.113	0.093	0.189	0.045	0.156	0.175	0.211	0.283	0.302	0.491
Computer_screen	0.101	0.057	0.149	0.148	**0.202**	0.096	0.053	0.078	0.058	0.114	0.087	0.092	0.105	0.108	0.106	0.053	0.092	0.054	0.116	0.110	0.114	0.140	0.142	0.202
Flag_US	0.227	0.201	0.239	0.233	0.232	0.185	0.163	0.088	0.092	0.212	0.222	0.252	0.098	0.096	0.217	0.151	0.104	0.063	0.199	0.278	0.274	0.337	**0.340**	0.340
Airplane	0.073	0.049	0.085	0.076	**0.189**	0.081	0.083	0.038	0.053	0.092	0.104	0.113	0.074	0.073	0.058	0.057	0.042	0.042	0.060	0.068	0.072	0.097	0.101	0.189
Car	0.252	0.237	0.298	0.297	**0.316**	0.232	0.204	0.146	0.188	0.250	0.272	0.286	0.247	0.254	0.228	0.203	0.148	0.180	0.245	0.269	0.282	0.261	0.270	0.316
Bus	0.013	0.022	0.010	0.015	**0.048**	0.011	0.010	0.011	0.011	0.013	0.011	0.011	0.011	0.009	0.012	0.010	0.011	0.011	0.013	0.011	0.014	0.011	0.012	0.048
Truck	0.038	0.040	0.038	0.045	**0.069**	0.039	0.046	0.020	0.026	0.038	0.047	0.047	0.030	0.029	0.036	0.044	0.021	0.026	0.037	0.045	0.044	0.036	0.037	0.069
Boat	0.096	0.073	0.127	0.116	**0.280**	0.044	0.050	0.028	0.024	0.052	0.052	0.061	0.038	0.040	0.037	0.054	0.039	0.027	0.055	0.053	0.065	0.063	0.065	0.280
Walking_running	0.353	0.325	0.348	0.344	0.396	0.370	0.370	0.305	0.321	0.377	0.403	0.418	0.408	0.418	0.367	0.370	0.302	0.318	0.371	0.402	0.412	0.410	**0.419**	0.419
People_marching	0.228	0.264	0.321	0.279	0.300	0.261	0.275	0.191	0.246	0.252	0.309	0.289	0.255	0.255	0.279	0.275	0.224	0.269	0.287	0.316	0.321	0.349	**0.353**	0.353
Explosion	0.098	0.094	0.154	**0.186**	0.182	0.082	0.094	0.024	0.034	0.092	0.152	0.143	0.021	0.049	0.069	0.077	0.028	0.031	0.089	0.100	0.113	0.078	0.089	0.186
Natural_disaster	0.055	0.049	**0.078**	0.030	0.061	0.049	0.034	0.017	0.033	0.055	0.055	0.056	0.036	0.038	0.038	0.030	0.015	0.023	0.043	0.042	0.042	0.039	0.043	0.078
Maps	0.476	0.610	0.673	0.684	**0.754**	0.380	0.598	0.146	0.142	0.420	0.635	0.659	0.009	0.017	0.366	0.536	0.132	0.142	0.397	0.625	0.631	0.648	0.661	0.754
Charts	0.327	0.392	0.523	**0.528**	0.520	0.295	0.277	0.087	0.179	0.313	0.433	0.412	0.124	0.142	0.215	0.320	0.148	0.067	0.217	0.436	0.438	0.519	0.527	0.528
MAP	**0.250**	**0.269**	**0.307**	**0.312**	**0.352**	**0.250**	**0.258**	**0.191**	**0.188**	**0.262**	**0.300**	**0.304**	**0.260**	**0.261**	**0.239**	**0.252**	**0.198**	**0.180**	**0.249**	**0.296**	**0.302**	**0.315**	**0.319**	**0.370**

Figure 3: Overview of our visual-only analysis experiments on the MediaMill Challenge using the feature vectors {**W1**,**W**,**G**,**S**}, classifiers {SVM, logistic regression, Fisher linear discriminant}, and classifiers settings {regular cross-validation, episode-constrained cross-validation (+), and inclusion of γ SVM parameter (g)} as explained in Section 2.3. The best result per concept is denoted in bold (data for 62 remaining Challenge concepts not shown).

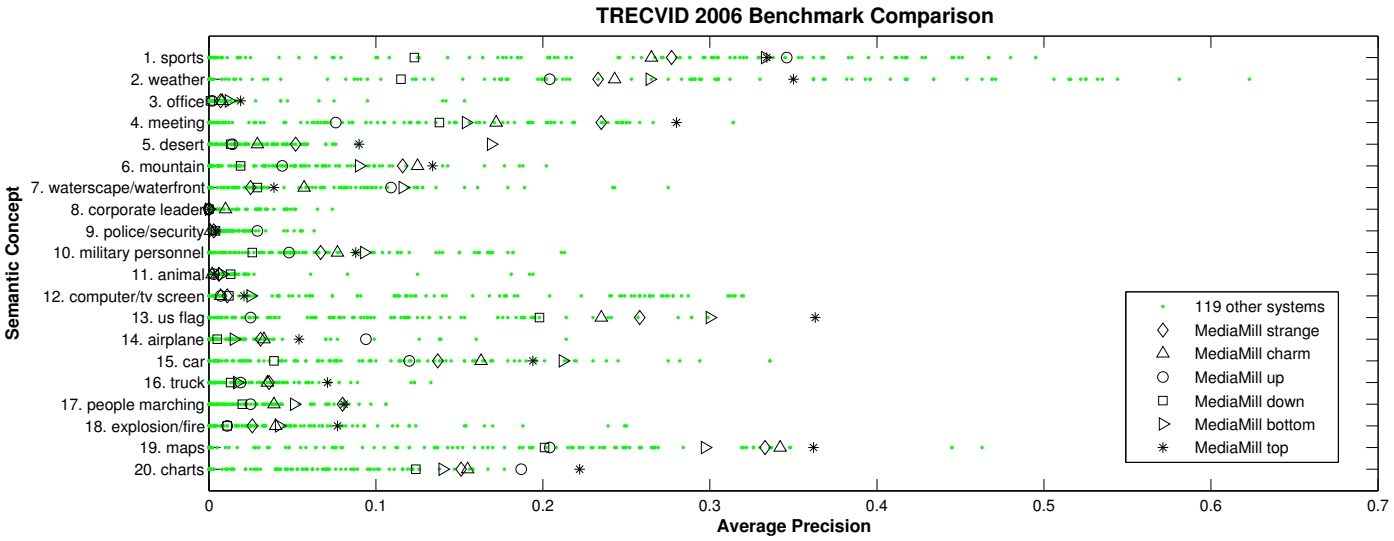

Figure 4: Comparison of MediaMill video indexing experiments with present-day indexing systems in the TRECVID 2006 benchmark.

2.5.1 Run 'strange': Best Visual-Only

Concept detection that relies on a single feature/classifier combination seldom leads to excellent performance. For some concepts, however, performance is reasonable, e.g., *meeting*, *desert*, *mountain*, *us flag*, *people marching*, *maps*, and *charts*. Our other runs more or less extend on this run to see how performance is influenced by: using concepts in context, adding text, comparison against a keypoint-only run, using cluster-based similarity, and late fusion of several visual-only analysis methods.

2.5.2 Run 'charm': Visual Context Analysis

The context analysis step adds context to our interpretation of the video. Here we combine the best visual-only concept analysis method per concept from Fig. 3, e.g., for *building* we use WGS features in combination with the Fisher linear discriminant, where for a *court* we use the WG features in combination with an SVM. The best visual-only run yields a probability for each shot and all 101 concepts detectors in our thesaurus. The probability indicates whether a concept is present. We fuse these probability scores into a concatenated vector for each shot. This vector forms the input for a stacked SVM. We use this vector to learn a classifier for each concept. We learn these concepts on Challenge validation set $\mathcal{B}$ to prevent overfitting in our context analysis step.

The results do not show a clear overall advantage of using context for concept detection. For concepts as *mountain*, *corporate leader*, and *military personnel*, context improves upon the best visual-only run. Context aids especially to disambiguate between *maps* and *charts*. For the other concepts the benefit of context is less apparent, but this might be caused by the fact that validation set $\mathcal{B}$ contains less examples than training set $\mathcal{A}$.

2.5.3 Run 'up': Early Fusion

For the 'up' run we performed an early fusion scheme similar to Challenge experiment 3. We combine the feature vectors resulting from visual feature extraction with those obtained from textual feature extraction using vector concatenation.

For the visual features we selected the **WG** combination. To obtain text features, we transformed the ASR text in three ways. The first transformation was pure normalization, eliminating punctuation and capitalization. The second transformation was stemming, using the Porter [25] stemmer to reduce the number of morphological variants of words. The third transformation of the text was character 4-grams, using consecutive sequences of 4 characters for search, to catch 'sounds-like' errors made by the speech transcriber. We used relevance feedback to select the most descriptive n terms for each concept. We did this by calculating Rocchio's weight for all of the terms of the ASR text of the positive concept examples, as described in [30]. We then selected the n terms with the highest weight. We experimented with several values for n, i.e. 50, 100, 200, and 500. For feature extraction we compare the text associated with a shot with the top n terms for a concept. This comparison yields a text vector for a shot, which contains the histogram of the words in association with a concept. To learn semantic concepts this text vector serves as the input for a logistic regression classifier. We did not found any significant differences between the text-only experiments, eventually we selected the unstemmed relevance feedback method with top *200* terms. The concatenation of the visual and textual features forms the input for an SVM, which learns detectors for all 101 concepts.

Early fusion performs reasonably well for *sports*, *waterscape*, *police/security*, *airplane*, and *charts*. Apparently, the text complements the visual features for these concepts. However, for the other concepts addition of text has a neg-

ative influence on concept detection performance. In such cases as *meeting*, *desert*, *mountain*, *us flag*, and *maps* resulting in poor performance when compared to our best results for these concepts. Early fusion suffers from textual features based on poor quality (machine translated) ASR.

2.5.4 Run 'down': Late Fusion of Keypoint Detectors

SIFT only operates on intensity images, ignoring color information. Van de Weijer [43] proposes a hue descriptor to be used in conjunction with SIFT. The hue descriptor is a 37-bin histogram of the hue of pixels in the interest region, weighed by saturation and the distance to the center of the interest region.

Besides adding color information in description, we can also add color in detection of interest regions. By applying color boosting [41] to the input image and extending Harris-Laplace to operate on color [31, 42], we can detect regions with strong color information. We will refer to this detector as Boosted ColorHarris-Laplace.

We have selected 5 combinations of detectors and descriptors based on experiments with the MediaMill Challenge:

- Harris-Laplace, SIFT
- Harris-Laplace, Hue *and* SIFT
- Boosted ColorHarris-Laplace, SIFT
- Boosted ColorHarris-Laplace, Hue
- Harris-Laplace *and* Boosted ColorHarris-Laplace, SIFT

For each of the five combinations of interest region detectors and descriptors we have applied SVM, yielding five ranked lists of shots. Shots in the list have a likelihood (provided by the SVM) and naturally the shots with the highest likelihood are ranked at the top. For late fusion of such ranked lists several methods exist, e.g., min, max, sum, median, and product [9]. An extension of product fusion that is capable to handle missing data is the geometric mean. We found after several experiments on 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. 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.

Visual inspection of results shows that there are many topics where many top ranked results do not look like the target concept at all (from a human perspective, at least). However, there is a pattern in those results: they all tend to have many smooth areas, be relatively blurred and/or lack saturated colors. These are all conditions in which an interest region detector will detect few interest regions. Looking at the results of our Harris-Laplace interest region detector we can see that there are many keyframes with few interest regions in this run. For the top 100 shots of the runs of all concepts evaluated this year, 30% have 10 interest regions or less in run 5. In all other runs it does not exceed 10%. One might be tempted to remove shots with few interest regions because they introduce many incorrect results, but this can have side-effects. For the first 100 shots of the concept animal, 10 shots have been evaluated as correct. However, five of these have less than 10 interest regions. Removing these shots would cause a serious decrease in performance for this concept. We are currently investigating how to handle keyframes with few detected interest points.

2.5.5 Run 'bottom': Proto-Concept Clustering

In contrast to using semantic proto-concepts as a vocabulary, we use a data-driven clustering approach to finding representative proto-concepts in run 'bottom'. A popular clustering approach for finding proto-concepts is k-means [6, 26, 38]. K-means is an unsupervised clustering algorithm that tries to minimizes the variance between k clusters and the training data, where k is a parameter of the algorithm. The advantages of k-means are its simple and efficient implementation. However, the disadvantage of k-means is that the algorithm is variance-based. Thus, the algorithm will award more clusters to high-frequency areas of the data, leaving less clusters for the remaining areas. This over-sampling of dense regions is unwanted, since frequent occurring data is not informative. In contrast to variance-based clustering, the prototypes for a codebook model are better represented by using radius-based clustering [18]. Radius-based clustering assigns all data points within a fixed radius of similarity r to one cluster, where r is a parameter of the algorithm. This radius r, denotes the maximum similarity between data points that may be considered equal. As such, the radius determines whether two patches describe the same prototype. The radius-based clustering algorithm we use is developed by Astrahan [2].

This run constructs a dictionary of proto-concepts for the Weibull and Gabor features in a data-driven approach. This data-driven approach was developed in parallel to the other experiments. Hence, this run is not incorporated in the fusion, best-visual or context runs. Nevertheless, the data-driven approach outperforms the other MediaMill runs for 6 out of 20 concepts. Moreover, the concept *desert* yields the best result over all other systems. Hence, a data-driven approach for finding a dictionary of proto-concepts complements the other runs and even yields first-rate performance for some concepts.

2.5.6 Run 'top': Late Fusion of Visual-Only Analysis

This run is a late fusion of all our experiments based on visual features. For the 39 TRECVID concepts all experiments from Fig. 3 and the keypoint feature run ('down') are included. However, fusing *all* experiments did not yield good results on Challenge data. Instead, we choose to use a variable number of experiments per concept. The combination always includes the keypoint feature run as an experiment. The combination method adds further experiments from Fig. 3 on a per-concept basis. Experiments are added in order of decreasing performance. We consider combinations of up to 10 experiments. Per concept we select the number of experiments that yields the best average precision performance on Challenge validation set $\mathcal{B}$. 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 best overall run. Moreover, we obtain the highest performance for pure visual concepts *flag us* and *charts*. We also perform well for concepts *meeting*, *desert*, and *maps*. For concepts with relatively few learning examples, e.g., *corporate leader* and *police/security*, classification remains hard. Relative to other concept detection methods we perform poor for *computer/tv screen*. This is caused, however, by the fact that we do not consider screens that appear in a news studio setting as valid examples. Since detection here boils down to detecting the studio or news anchor. It is interesting to note that fusion always outperforms the best single visual-only analysis approach, except for *animal* where both scores are close to zero. The 'bottom' run was not included in the fusion, inclusion of this run in the fusion will further improve concept classification performance.

2.6 Scaling-up to 491 Concept Detectors

To scale our lexicon of concept detectors further we adopt a graceful degradation approach. For the remaining 62 MediaMill concepts, the keypoint features from the 'down' run and the SVM gamma experiment are not available. We determine the best combination of experiments for these concepts from the remaining experiments; again up to 10 experiments are allowed in a combination. For the LSCOM concepts [23] none of the SVM experiments are available, leading to a further reduction in the number of experiments, i.e. only those performed by logistic regression and Fisher's linear discriminant. Because parameter optimization of the SVM is expensive – even when supercomputers are used – performing a complete analysis for all concepts was not feasible. While the performance might not be optimal, the detectors may still be useful for semantic video retrieval.

3 Semantic Video Retrieval

Our TRECVID 2006 search task efforts have concentrated on automatic and interactive retrieval using the lexicon of 491 learned concept detectors. For users, remembering a list of 491 concepts is not feasible. We therefore developed a concept suggestion engine which finds the most appropriate concept detector given the topic using an ontology. Query by this concept yields a ranking of the data, a convenient way of browsing the result is our CrossBrowser [36] 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 RotorBrowser which allows the user to browse along up to 8 directions. In addition to concept suggestion and video browsing, we also explored the benefit of query-by-example using objects in images rather than descriptors based on global characteristics of the images.

3.1 Automatic Search: Concept Suggestion

Selection of a concept detector appropriate to the query can allow users to quickly retrieve a list of relevant video fragments. We focus on the selection of a single best detector to maximize retrieval performance. We used the automatic search task to evaluate two different techniques for selecting the single best detector for a particular topic, namely: text matching and ontology querying. We compare results with a text retrieval baseline.

The text baseline this year was created using the Lucene [1] search engine. The final baseline was the result of the combination of search results of a number of different searches using different types of queries. These searches made use of the transcriptions and the machine translations that were provided, as well as the story boundaries supplied by the DVMM lab at Columbia University [16]. Speech was indexed at two temporal levels: shot level and story level. By indexing at story level we adjust for the temporal mismatch between the time a visual object is mentioned in speech and the time it appears in a shot. We used different forms of text normalization to increase recall. Story-level speech was also used to increase recall. As we did last year, we boosted precision by performing an extra search on proper nouns for specific queries, and an extra search for all nouns for general queries [34] The final baseline result was created by combining different searches. The text transformations we used are the following: *character 4-grams, Porter [25] stemmed*, *exact match*, *proper nouns/all nouns*. Each search was performed at shot-level text and on story-level text. Finally all searches were combined using Borda fusion, as preliminary experiments showed that the highest MAP was obtained in this way.

3.1.1 Detector Selection through Text Matching

Our text matching detector selection technique is based on statistical text retrieval of the detector description that best matches the query text. An in-depth description can be found in [33]. We index the concept descriptions that are given to annotators creating the ground truth for the topics,

Table 1: Comparison of two detector selection strategies for video retrieval. Search results are compared against a text only baseline. The best result is given in bold. Note that for topic 0194 no suitable detector was found, we used a simple default text search instead.

Search Topic		Baseline	Text Match		Ontology Querying	
ID	**Query**	**AP**	**Selected Detector**	**AP**	**Selected Detector**	**AP**
0173	emergency vehicles in motion	**0.007**	Emergency Vehicle	0.006	Emergency Vehicles	0.006
0174	tall buildings and the top story visible	0.001	Cityscape	**0.023**	Cityscape	**0.023**
0175	people leaving or entering a vehicle	**0.001**	Vehicle	**0.001**	Rowboat	0.000
0176	soldiers, police, or guards escorting a prisoner	**0.005**	Guard	0.001	Prisoner	0.000
0177	daytime demonstration or protest with building visible	**0.046**	Demonstration Or Protest	0.014	Singing	0.000
0178	US Vice President Dick Cheney	**0.252**	Emile Lahoud	0.000	Corporate Leader	0.000
0179	Saddam Hussein with another persons face visible	**0.134**	Person	0.000	Face	0.000
0180	people in uniform and in formation	0.001	Insurgents	**0.002**	Ties	0.000
0181	US President George W. Bush, Jr. walking	**0.028**	George Bush jr	0.012	George Bush jr	0.012
0182	soldiers or police with weapons and military vehicles	**0.054**	Emergency Vehicles	0.000	Tanks	0.008
0183	water with boats or ships	**0.028**	Ship	0.003	Rowboat	0.000
0184	people seated at a computer with display visible	**0.005**	Furniture	0.000	Computers	0.004
0185	people reading a newspaper	0.007	Newspaper	**0.084**	Newspaper	**0.084**
0186	a natural scene	0.004	Beach	**0.014**	Waterfall	0.003
0187	helicopters in flight	0.011	Helicopters	**0.016**	Airplane Takeoff	0.001
0188	something burning with flames visible	**0.116**	Coal Powerplants	0.000	Sitting	0.000
0189	people dressed in suits, seated, and with flag	0.000	Group	**0.009**	Flag USA	0.001
0190	at least one person and at least 10 books	**0.005**	Single Person	0.000	Graphical Map	0.000
0191	at least one adult person and at least one child	**0.008**	Adult	0.001	Infants	0.001
0192	a greeting by at least one kiss on the cheek	**0.001**	Election Greeting	0.000	Election Greeting	0.000
0193	smokestacks, chimneys, or cooling towers with smoke	0.000	Smoke Stack	**0.009**	Smoke Stack	**0.009**
0194	Condoleeza Rice	**0.141**	-	0.127	Sky	0.000
0195	soccer goalposts	0.139	Soccer Game	**0.608**	Baseball Game	0.000
0196	scenes with snow	**0.165**	Snow	0.122	Snow	0.122
	MAP	*0.048*		*0.044*		*0.011*

once again using the Lucene search engine. Each description elaborates on the visual elements that should — or should not — be present in a shot for it to be considered relevant. For example, the description for our concept detector *storms* is 'outdoor scenes of stormy weather, thunderstorms, lightning.' It explicitly indicates that video containing lightning and thunderstorms should be tagged as storms. The descriptions are by no means exhaustive, usually consisting of one or two sentences [23, 37], but do contain a significant amount of information about the different kinds of visual content associated with each detector. By matching the query to detector description, we hope to select the detector that best matches the description.

3.1.2 Detector Selection through Ontology Querying

In an attempt to model the user intent, we design a detector selection method based on ontology querying. This detector selection method is described in [33], which can be summarized as follows: We first link each detector to a noun synset (or particular meaning) in the WordNet [7] ontology. At runtime we use a memory-based shallow parser, described in [39], to extract nouns and noun chunks from the topic text. We then look up each noun chunk in WordNet. When a match has been found the matched words are eliminated from further lookups. Then, we look up any remaining nouns in WordNet. The result is a number of WordNet nouns related to the query text. Each WordNet noun can have several different synsets. We reduce each noun to its most common synset, as this form of disambiguation has been shown to work adequately in the past [17] We calculate the similarity of each of the topic synsets to each of the detector synsets using Resnik's measure [28], where a concept is viewed as the composite of its synonyms and its sub-concepts.

3.1.3 Submitted Automatic Search Results

Likely due to our choice to focus on the detection of a single concept detector, rather than combining multiple information sources (text, low-level features, more detectors, and so on) our automatic search results (shown in Table 1) did not stand out of the crowd. Surprisingly, of the three automatic runs we entered this year, the run with the highest mean average precision was the text baseline, with a mean average precision of 0.048. The single best detector based on text matching performed almost as well as the text baseline, with a mean average precision of 0.044. The single best ontology query based detector had a much lower mean average precision of 0.0114. We expect that the good performance of the baseline is largely due to the inclusion of story-level ASR and boosting nouns. The baseline always performed better than the median for topics containing proper nouns (for example, 'Dick Cheney' and 'Saddam Hussein'). Furthermore, we notice that the detector match tends to perform better than the baseline for topics that request incidental visual objects such as the *newspaper* detector used for 'one or more people reading a newspaper' and the *tie* detector selected for 'multiple people in uniform and formation'.

3.2 Interactive Search: Video Browsing

In traditional video retrieval systems a user typically creates a search query, then 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 by providing users with query by object matching and browsers that allow to visualize the entire data set in multiple dimensions and facilitate interactive exploration.

3.2.1 Query by Object Matching

In order to interactively search for specific objects and locations using their visual properties we implemented a *query by object matching* algorithm. The method of object matching draws heavily on the work of Sivic and Zisserman [32] and can be summarized as follows. Firstly, interest points are detected and described in each keyframe using the methods of Lowe [19] i.e. we use a Difference of Gaussians detector and the Scale Invariant Feature Transform (SIFT) descriptor, no color information was used. Next, SIFT vectors were quantized into a fixed set of 10,000 prototypes by applying a clustering algorithm based on competitive learning. Lastly, text-retrieval methods are applied using a term-document matrix (in our case a prototype-keyframe matrix) and td-idf weighting. In practice, the query procedure starts with a keyframe displayed in the GUI, the user then selects a region by drawing a rectangle around the object of interest and a ranked list of keyframes containing similar groups of SIFT prototypes is returned.

3.2.2 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. We define several thread types in our system. The most used form of threads is the query result thread: the result of a user constructed query. In this case the shots are dynamically linked because they originate from the same query result. Other forms of threads include visual threads, semantic threads, top-rank threads, textual threads, and the time thread. The visual thread links shots together which share the same visual characteristics, so that shots next to each other are also visually similar. The semantic thread links shots together based on their detected concept scores, so that shots next to each share common semantics. The textual thread links shots to each other in which similar words are spoken, derived from the ASR text. The time thread follows the time line of a video. The top-rank thread connects the top n shots from every concept.

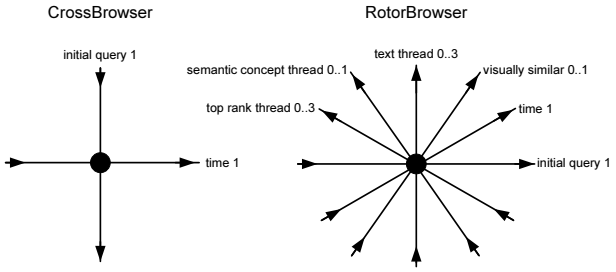

Figure 5: Browsing dimensions for the CrossBrowser and RotorBrowser [29]. The number behind each dimension indicates the number of times this dimension can be shown for any shot.

3.2.3 Thread Visualizations

The MediaMill engine supports two modes for displaying threads. Both show an active focal shot and a collection of threads relevant to this focal shot. The display modes use a fixed layout. The current focal point is displayed with the largest keyframe and is centered on the screen. All relevant threads are shown in a star formation around it. The user has to choose between two actions only: selection of the current focal shot as a valid result, or switch focus to any of the neighboring shots in one of the threads. In addition, the user may use the mouse to directly select any visible shot by clicking on the key frame representing it.

To determine the benefit of having additional dimensions, one display mode, i.e. the CrossBrowser, is limited to showing two fixed directions only. Namely, the query result thread and the time thread. The other display mode, i.e. the multi-dimensional RotorBrowser [29], shows a variable number of directions. See Fig. 5 for an overview of available directions. Depending on the thread type, the system may show a thread once or multiple times. For example: a shot can only participate once in the time thread, the visually similar thread and the semantic concept thread. Multiple concepts can be detected from a shot, however, so this could lead to inclusion of the shot in multiple top-rank threads. The same holds for textual threads: the shot could be a result of multiple textual queries, so these could all be shown. We limit the number of visible threads to reduce the amount of information a user has to process. To achieve this, the RotorBrowser uses a priority ranking system where the initial query result and the time thread are placed first, followed by visually similar, text, semantic concept and finally up to three top-rank threads.

The CrossBrowser allows movement through the initial query results, and for each retrieved shot limited movement through the time thread. To preserve context the user is not allowed to leave the initial query results, except when a new query is posed to the system. Because of these limitations in browsing possibilities the CrossBrowser works well for focussed topics. In contrast, the RotorBrowser shows all possible relevant threads for each shot. Moreover, the RotorBrowser does allow the user to leave the initial query result set. By doing so the user can browse through any-

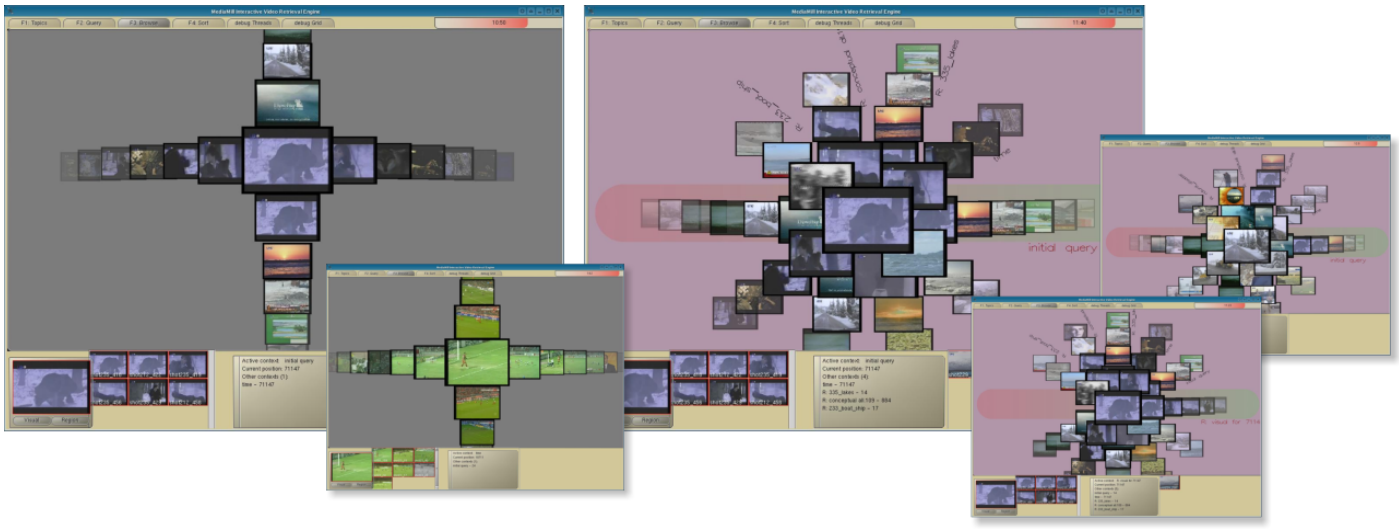

Figure 6: Screenshots of the CrossBrowser (left) and the RotorBrowser.

thing that catches her interest. To prevent the user from "getting lost" the user can always jump back to the latest query result. Hence, the RotorBrowser allows to explore all possible directions for complicated topics. A graphical overview of the browsers is depicted in Fig. 6.

3.2.4 Submitted Interactive Search Results

We submitted three runs for interactive search with three expert users. One user performed the interactive search by using the MediaMill search engine with the CrossBrowser. Another user exploited the MediaMill system in combination with the RotorBrowser. We dedicated one run to evaluating the object matching functionality. During this run, the following procedure was used. Firstly, any of the other interactive search tools available in the system was used to generate a first set of keyframes to use as entry points for object matching. Next, object matching queries were attempted using the keyframe entry points. The user tried to use the object-matching functionality whenever possible, but if this was judged to be hopeless the search was completed using the alternative tools. Results in Fig. 7 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 14 out of 24 topics. Best performance is obtained for 6 topics. Overall the user of the CrossBrowser obtains better performance than the user of the RotorBrowser and the user of the query by object matching functionality.

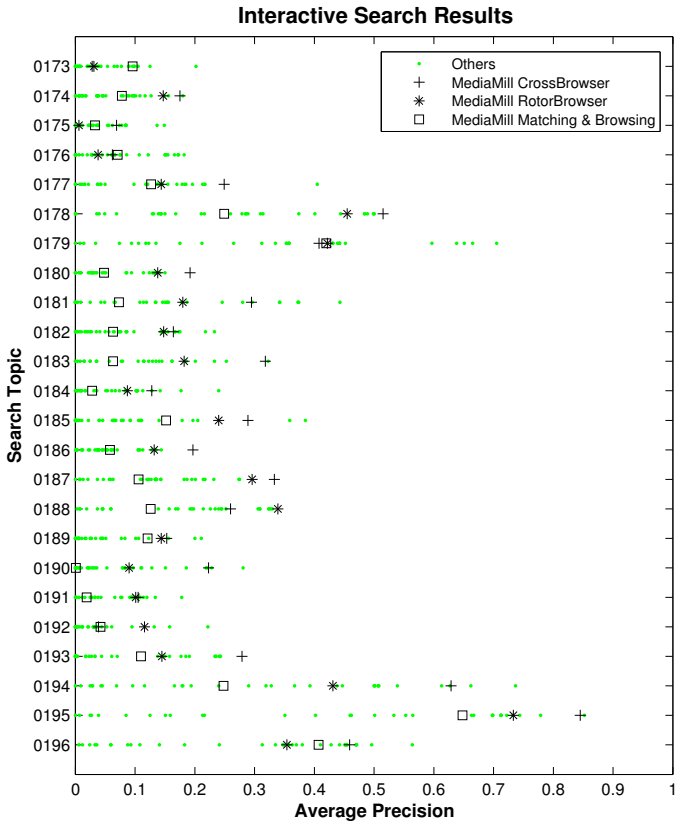

Figure 7: Comparison of interactive video search results for 24 topics (Table 1) performed by 36 users of present-day video retrieval systems. MediaMill results are indicated with special markers.

Among the TRECVID 2006 interactive search topics there are few topics that can be successfully approached by object matching. For most search topics object matching was therefore abandoned early in the search and in the end it turned out to be useful for only 2 of the 20 topic results; topic 0179 (Saddam Hussein with another person's face visible) and topic 0185 (people reading a newspaper). For topic 0179 an image of a courtroom was used as an entry point and other images of the same courtroom were retrieved. For topic 0185 an image of a television studio was used as an entry point and other images of the same studio were retrieved (the studio was the backdrop of a program that discussed news stories and the broadcast sometimes

showed people reading a newspaper story). These results suggest that, although object matching is not suitable for the majority of TRECVID 2006 topics, when a topic can be related to a specific object or location it can produce competitive results.

The CrossBrowser is especially successful when a search topic can be addressed with a single concept detector from the lexicon. Finding a helicopter in flight, for example, is relatively easy when a reasonably accurate *helicopter* detector is available. The CrossBrowser then allows for quick scanning and selection of relevant results. This observation also holds for topics 0174, 0177, 0180, 0183, 0185, 0186, 0195. Although we have the best score for the topic requesting shots from Dick Cheney, we did not have a specific Dick Cheney detector available. Because the (visual) appearance of specific persons in broadcast news is often described in the speech signal of an anchor or voice-over, for example, query-by-keyword is effective for this topic. Once a relevant shot is retrieved, the time tread aids in further augmentation of correct results. When search topics contain combinations of several reliable concept detectors, e.g. *people*, *suits*, *flag* (Topic: 0189), results are not optimal. This indicates that much is to be expected from a more intelligent combination of query results. Overall the CrossBrowser ranks 2nd among all interactive video retrieval systems this year.

Overall the RotorBrowser performed well in the TRECVID evaluation, ranked 6th in the overall results. When compared to the CrossBrowser there is however a significant gap in results. Unfortunately despite the fact that both runs were performed by an expert user for each respective browser, this does not make the results themselves comparable, so we cannot determine which browser is best. A more detailed user study is required before we can answer this question. We can however deduce some interesting facts from the results.

Analysis of our topic evaluation runs shows that the RotorBrowser required less user interaction than the CrossBrowser. If we count every keyboard press or mouse click as one move action, the RotorBrowser required 1491 moves per topic, and the CrossBrowser required 1829 moves per topic, on average. More interesting insights stem from analyzing the various threads used for retrieval with the RotorBrowser, depicted in Fig. 8. By far the largest portion of results is generated from the initial query results and the time thread. Since these threads are the only two threads available in the CrossBrowser this partly explains its success.

A quarter of the selected shots was generated by using mouse interaction. However from this part only 41.9% of the shots were judged correct by TRECVID. This contrasts the initial query results and the time thread, where approx. 61.0% was judged relevant. The cause lies in the fact that the mouse was only used when the user saw a possibly valid shot on the visible edge of some thread on the screen. If the shot was closer to the center the keyboard would have been used, since it allowed for quicker navigation. The fixed layout of the RotorBrowser also resulted in these shots being the smallest on screen, which could account for the user making the most mistakes in determining if a shot was correct.

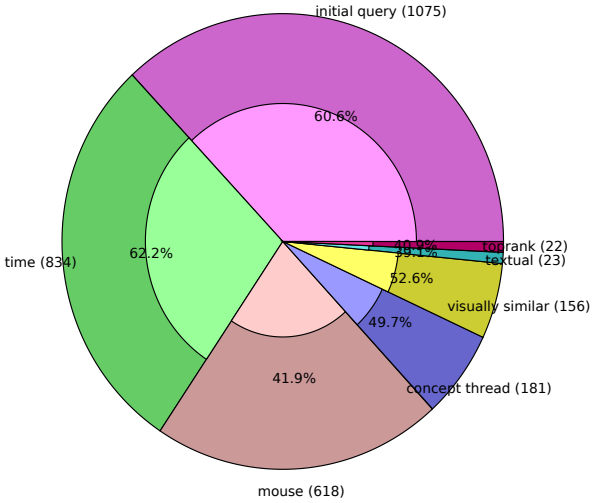

Figure 8: This graph shows how many results were obtained from each thread, and which percentage of them was judged relevant.

From the other threads the semantic concept thread and the visually similar thread were used most. Since the textual threads were visible only when the user explicitly performed a textual search these were seldom used. Although the top-rank thread occupies the largest part of the screen these are also hardly used by the user of the RotorBrowser. Since it does require a lot of mental processing by the user, the results suggest that the top-rank threads should be shown on user request only. Overall we can say that the RotorBrowser is able to find similar results as the CrossBrowser with less user interaction. However, more tuning is required to make it visualize relevant threads only. A more in-depth study using a larger (novice) user base is currently underway to determine the possible benefit of having multiple dimensions in browsing.

4 Lessons Learned

TRECVID continues to be a rewarding experience in gaining insight in the difficult problem of semantic video indexing and retrieval. To conclude this paper we highlight our most important lessons learned:

- *The MediaMill Challenge allows to gain insight in intermediate video analysis steps by fostering repeatability of experiments on system components;*
- *Regional image features seem more effective for concept detection than global or keypoint features;*
- *Data-driven clustering of proto-concepts is more effective than using similarity to a predefined set;*

- *Coloring keypoint features matters;*
- *Keypoint methods become unstable when there are only few interest regions;*
- *A combination of visual analysis methods pays off;*
- *Simple classifiers can yield competitive performance;*
- *High dimensional features spaces can be mapped to semantic concepts using relatively simple classifiers like Fisher's linear discriminant;*
- *Learning concept detectors from few examples remains problematic;*
- *Early fusion of the textual modality and the visual modality helps for some concepts, but often yields decrease in performance due to modest quality of non-English speech recognition and machine translations;*
- *Late fusion of concept detectors using geometric mean is cheap and effective;*
- *Usage of supercomputers is seriously hampered by lack of efficient and effective data management tools;*
- *Scaling-up to 1,000+ detectors is a matter of annotated examples;*
- *Concept suggestion is a useful tool for video search engines containing 100+ detectors;*
- *Text matching and ontology querying are effective techniques for concept suggestion;*
- *Standard text retrieval is, in general, more effective than retrieval based on any single concept detector;*
- *Combining concept detectors effectively for video retrieval is an unsolved problem;*
- *Querying video archives by matching objects in keyframes is helpful for specific topics;*
- *Using multiple dimensions for browsing reduces the amount of user interaction;*

Acknowledgments

This research is sponsored by the BSIK MultimediaN project and the NWO MuNCH project. The authors are grateful to NIST and the TRECVID coordinators for the benchmark organization effort. We thank Cees Sterks and LOFAR for providing access to the Stella supercomputer.

References

[1] The Lucene search engine, 2006. `http://lucene.apache.org/`.

[2] M. Astrahan. Speech analysis by clustering or the hyperphoneme method, 1970. Stanford A.I. Project Memo, Stanford University, CA.

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

[4] 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/`.

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

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

[7] C. Fellbaum, editor. *WordNet: an electronic lexical database*. The MIT Press, Cambridge, USA, 1998.

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

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

[10] J. Gauvain, L. Lamel, and G. Adda. The LIMSI broadcast news transcription system. *Speech Communication*, 37(1–2):89–108, 2002.

[11] 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. Workshop Semantic Learning Applications in Multimedia, in conjunction with CVPR'06*, New York, USA, June 2006.

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

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

[14] J.M. Geusebroek. Compact object descriptors from local colour invariant histograms. In *British Machine Vision Conf.*, Edinburgh, UK, September 2006.

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

[16] W. H. Hsu and S.-F. Chang. Visual cue cluster construction via information bottleneck principle and kernel density estimation. In *The 4th Int. Conf. on Image and Video Retrieval*, Singapore, 2005.

[17] B. Huurnink. AutoSeek: Towards a fully automated video search system. Master's thesis, University of Amsterdam, October 2005.

[18] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, pages 604–610, 2005.

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

[20] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.

[21] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

[22] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

[23] 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.

[24] J. Platt. Probabilities for SV machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

[25] M. F. Porter. An algorithm for suffix stripping. In *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann, San Francisco, CA, 1997.

[26] 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. Conf. on Computer Vision*, 2005.

[27] G. Quénot, D. Moraru, L. Besacier, and P. Mulhem. CLIPS at TREC-11: Experiments in video retrieval. In E. Voorhees and L. Buckland, editors, *Proceedings of the 11th Text REtrieval Conf.*, volume 500-251 of *NIST Special Publication*, Gaithersburg, USA, 2002.

[28] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Int. Joint Conf. Artificial Intelligence*, San Mateo, CA, 1995.

[29] O. de Rooij, C. G. M. Snoek and M. Worring. Multi Thread Video Browsing. In *Int. Conf. Multimedia & Expo*, Beijing, China 2007. *Submitted*.

[30] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. In *Readings in Information Retrieval*, pages 355–364. Morgan Kaufmann, San Francisco, CA, 1997.

[31] K. E. A. van de Sande. Coloring concept detection in video using interest regions. Master's thesis, Universiteit van Amsterdam, 2007.

[32] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.

[33] 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*, 2007. *Submitted*.

[34] C. G. M. Snoek, J. C. van Gemert, J.M. Geusebroek, B. Huurnink, D. C. Koelma, G. P. Nguyen, O. de Rooij, F. J. Seinstra, A. W. M. Smeulders, C. J. Veenman, and M. Worring. The MediaMill TRECVID 2005 semantic video search engine, November 2005.

[35] 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 Anal. Mach. Intell.*, 28(10):1678–1689, October 2006.

[36] 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, February 2007.

[37] 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 *ACM Multimedia*, pages 421–430, Santa Barbara, USA, October 2006.

[38] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1299–1306. MIT Press, Cambridge, MA, 2006.

[39] E. F. Tjong Kim Sang. Memory-based shallow parsing. *Journal of Machine Learning Research*, 2:559–594, 2002.

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

[41] J. van de Weijer and Th. Gevers. Boosting saliency in color image features. In C. Schmid, S. Soatto, and C. Tomasi, editors, *IEEE Conf. on Computer Vision & Pattern Recognition*, June 2005.

[42] J. van de Weijer, Th. Gevers, and J.M. Geusebroek. Edge and corner detection by photometric quasi-invariants. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):625–630, 2005.

[43] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *European Conf. Computer Vision*, volume Part II, pages 334–348. Springer, 2006.

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

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

[46] H. Wactlar, M. Christel, Y. Gong, and A. Hauptmann. Lessons learned from building a terabyte digital video library. *IEEE Computer*, 32(2):66–73, 1999.

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