The MediaMill TRECVID 2005 Semantic Video Search Engine

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, 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 2005 experiments. The UvA-MediaMill team participated in four tasks. For the detection of camera work (runid: A_CAM) we investigate the benefit of using a tessellation of detectors in combination with supervised learning over a standard approach using global image information. Experiments indicate that average precision results increase drastically, especially for pan (+51%) and tilt (+28%). For concept detection we propose a generic approach using our semantic pathfinder. Most important novelty compared to last years system is the improved visual analysis using proto-concepts based on Wiccest features. In addition, the path selection mechanism was extended. Based on the semantic pathfinder architecture we are currently able to detect an unprecedented lexicon of 101 semantic concepts in a generic fashion. We performed a large set of experiments (runid: B_vA). The results show that an optimal strategy for generic multimedia analysis is one that learns from the training set on a per-concept basis which tactic to follow. Experiments also indicate that our visual analysis approach is highly promising. The lexicon of 101 semantic concepts forms the basis for our search experiments (runid: B_2_A-MM). We participated in automatic, manual (using only visual information), and interactive search. The lexicon-driven retrieval paradigm aids substantially in all search tasks. When coupled with interaction, exploiting several novel browsing schemes of our semantic video search engine, results are excellent. We obtain a top-3 result for 19 out of 24 search topics. In addition, we obtain the highest mean average precision of all search participants. We exploited the technology developed for the above tasks to explore the BBC rushes. Most intriguing result is that from the lexicon of 101 visual-only models trained for news data 25 concepts perform reasonably well on BBC data also.

1 Introduction

Despite the emergence of commercial video search engines, such as Google [9] and Blinkx [3], multimedia retrieval is by no means a solved problem. In fact, present day video search engines rely mainly on text - in the form of closed captions [9] or transcribed speech [3] - for retrieval. 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 or The Netherlands, querying the content becomes even harder as automatic speech recognition results are much poorer. For videos from these sources, an additional visual analysis potentially yields more robustness. For effective video retrieval there is a need for multimedia analysis; in which text retrieval is an important factor, but not the decisive element. We advocate that the ideal multimedia retrieval system should first learn a large lexicon of concepts, based on multimedia analysis, to be used for the initial search. Then, the ideal system should employ similarity and interaction to refine the search until satisfaction.

We propose a multimedia retrieval paradigm built on three principles: learning of a lexicon of semantic concepts, multimedia data similarity, and user interaction. Within the proposed paradigm, we explore the combination of query-by-concept, query-by-similarity, and interactive filtering using advanced visualizations of the MediaMill semantic video search engine. To demonstrate the effectiveness of our multimedia retrieval paradigm, several components are evaluated within the 2005 NIST TRECVID video retrieval benchmark [16].

The organization of this paper is as follows. First, we discuss our general learning architecture and data preparation steps. Our system architecture for generic semantic indexing is presented in Section 3. We describe our approach for camera work indexing in Section 4. Our multimedia retrieval paradigm is presented in Section 5. Our explorative work on BBC rushes is addressed in Section 6.

2 Preliminaries

The MediaMill semantic video search engine exploits a common architecture with a standardized input-output model to allow for semantic integration. The conventions to describe the modular system architecture are indicated in Fig. 1.

2.1 General Learning Architecture

We perceive of video indexing as a pattern recognition problem. We first need to segment a video. We opt for camera shots [18], indicated by i, following the standard in TRECVID evaluations. Given pattern x, part of a shot,

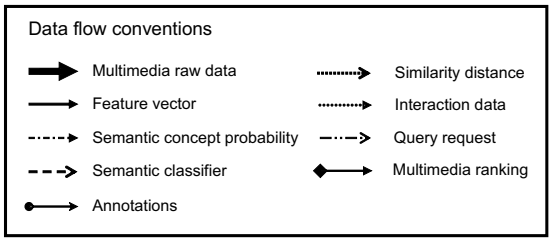

Figure 1: Data flow conventions as used in this paper. Different arrows indicate difference in data flows.

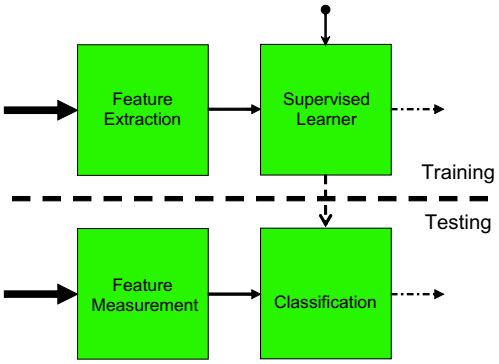

Figure 2: General architecture for supervised learning in in the MediaMill semantic video search engine, using the conventions of Fig. 1.

the aim is to detect an index ω from shot i using probability $p_i(\omega|x_i)$. We exploit supervised learning to learn the relation between ω and x_i. The training data of the multimedia archive, together with labeled samples, are for learning classifiers. The other data, the test data, are set aside for testing. The general architecture for supervised learning in the MediaMill semantic video search engine architecture is illustrated in Fig. 2.

We can choose from a large variety of supervised machine learning approaches to obtain $p_i(\omega|x_i)$. For our purpose, the method of choice should be capable of handling video documents. To that end, ideally it must learn from a limited number of examples, it must handle unbalanced data, and it should account for unknown or erroneously detected data. In such heavy demands, the Support Vector Machine (SVM) framework [35, 4] has proven to be a solid choice [1, 29]. The usual SVM method provides a margin in the result. We prefer Platt's conversion method [19] to achieve a posterior probability of the result. SVM classifiers thus trained for ω, result in an estimate $p_i(\omega|x_i, \vec{q})$, where $\vec{q}$ are parameters of the SVM yet to be optimized.

The influence of the SVM parameters on video indexing is significant [14]. We obtain good parameter settings for a classifier, by using an iterative search on a large number of SVM parameter combinations. We measure average precision performance of all parameter combinations and select the combination that yields the best performance, $\vec{q}^*$. Here we use 3-fold cross validation [11] with 3 repetitions to prevent overfitting of parameters. The result of the parameter search over $\vec{q}$ is the improved model $p_i^*(\omega|x_i, \vec{q}^*)$. In the following we drop $\vec{q}^*$ where obvious.

2.2 Data Preparation

Supervised learning requires labeled examples. In part, we rely on the provided ground truth of the TRECVID 2005 common annotation effort [36]. It is extended manually to arrive at an incomplete, but reliable ground truth for an unprecedented amount of 101 semantic concepts in lexicon Λ_S. In addition, we manually labeled a substantial part of the training set with respect to dominant type of camera work, i.e. *pan*, *tilt*, and/or *zoom* , if present.

In order to recognize concepts based on low-level visual analysis, we annotated 15 different proto-concepts: building (321), car (192), charts (52), crowd (270), desert (82), fire (67), US-flag (98), maps (44), mountain (41), road (143), sky (291), smoke (64), snow (24), vegetation (242), water (108), where the number in brackets indicates the number of annotation samples of that concept. We again used the TRECVID 2005 common annotation effort as a basis for selecting relevant shots containing the proto-concepts. In those shots, we annotated rectangular regions where the proto-concept is visible for at least 20 frames.

We split the training data a priori into four non-overlapping training and validation sets to prevent overfitting of classifiers. Training sets A, B, and C contain 30% percent of the 2005 training data, validation set D contains the remaining 10%. We assign all shots in the training set randomly to either set A, B, C, or D.

3 Semantic Pathfinder Indexing

The central assumption in our semantic indexing architecture is that any broadcast video is the result of an authoring process. When we want to extract semantics from a digital broadcast video this authoring process needs to be reversed. For authoring-driven analysis we proposed the semantic pathfinder [30]. The semantic pathfinder is composed of three analysis steps. It follows the reverse authoring process. Each analysis step in the path detects semantic concepts. In addition, one can exploit the output of an analysis step in the path as the input for the next one. The semantic pathfinder starts in the *content analysis step*. In this analysis step, we follow a data-driven approach of indexing semantics. The *style analysis step* is the second analysis step. Here we tackle the indexing problem by viewing a video from the perspective of production. This analysis step aids especially in indexing of rich semantics. Finally, to enhance the indexes further, in the *context analysis step*, we view semantics in context. One would expect that some concepts, like *vegetation*, have their emphasis on content where the style (of the camera work that is) and context (of

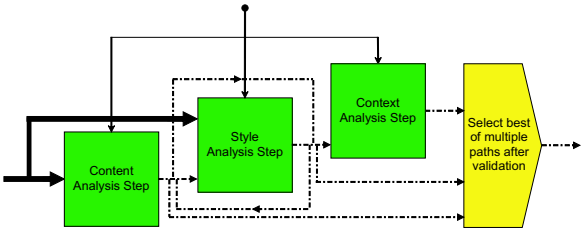

Figure 3: The semantic pathfinder for one concept, using the conventions of Fig. 1.

concepts like *graphics*) do not add much. In contrast, more complex events, like *people walking*, profit from incremental adaptation of the analysis to the intention of the author. The virtue of the semantic pathfinder is its ability to find the best path of analysis steps on a per-concept basis. An overview of the semantic pathfinder is given in Fig. 3.

3.1 Content Analysis Step

We view of video in the content analysis step from the data perspective. In general, three data streams or modalities exist in video, namely the auditory modality, the textual modality, and the visual one. As speech is often the most informative part of the auditory source, we focus on visual features, and on textual features obtained from transcribed speech. After modality specific data processing, we combine features in a multimodal representation using early fusion and late fusion [32].

3.1.1 Visual Analysis

Modeling visual data heavily relies on qualitative features. Good features describe the relevant information in an image while reducing the amount of data representing the image. To achieve this goal, we use Wiccest features as introduced in [6]. Wiccest features combine color invariance with natural image statistics. Color invariance aims to remove accidental lighting conditions, while natural image statistics efficiently represent image data.

Color invariance aims at keeping the measurements constant under varying intensity, viewpoint and shading. In [7] several color invariants are described. We use the $\mathcal{W}$ invariant that normalizes the spectral information with the energy. This normalization makes the measurements independent of illumination changes under uniform lighting conditions.

When modeling scenes, edges are highly informative. Edges reveal where one region ends and another begins. Thus, an edge has at least twice the information content then a uniformly colored patch, since an edge contains information about all regions it divides. Besides serving as region boundaries, an ensemble of edges describes texture information. Texture characterizes the material an object is made of. Moreover, a compilation of cluttered objects can

Figure 4: An example of dividing an image up in overlapping regions. In this particular example, the region size is a $\frac{1}{2}$ of the image size for both the x-dimension and y-dimension. The regions are uniformly sampled across the image with a step size of half a region. Sampling in this manner identifies nine overlapping regions.

be described as texture information. Therefore, a scene can be modeled with textured regions.

Texture is described by the distribution of edges at a certain region in an image. Hence, a histogram of a Gaussian derivative filters represents the edge statistics. Since there are more non-edge pixels then there are edge pixels, the distribution of edge responses for natural images always 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. This specific distribution can be well modeled with an integrated Weibull distribution [8]. This distribution is given by

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

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 origin of the distribution.

To assess the similarity between Wiccest features, a goodness-of-fit test is utilized. The measure is based on the integrated squared error between the two cumulative distributions, which is obtained by a Cramér-von Mises measure. For two Weibull distributions with parameters F_β, F_γ and G_β, G_γ a first order Taylor approximation of the Cramér-von Mises statistic yields the log difference between the parameters. Therefore, a measure of similarity between two Weibull distributions F and G is given by the ratio of the parameters,

$$W2(F,G) = \sqrt{\frac{\min(F_\beta,G_\beta)}{\max(F_\beta,G_\beta)}\frac{\min(F_\gamma,G_\gamma)}{\max(F_\gamma,G_\gamma)}} \quad . \tag{2}$$

The μ parameter represents 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 may be ignored.

In summary, Wiccest features provide a color invariant texture descriptor. Moreover, the features rely heavily on natural image statistics to compactly represent the visual information.

3.1.2 Contextures: Regional Texture Descriptors and their Context

The visual detectors aim to decompose an image in proto-concepts like vegetation, water, fire, sky etc. To achieve this goal, an image is divided up in several overlapping rectangular regions. The regions are uniformly sampled across the image, with a step size of half a region. The region size has to be large enough to assess statistical relevance, and small enough to capture local textures in an image. We utilize a multi-scale approach, using small and large regions. An example of region sampling is displayed in figure 4.

A visual scene is characterized by both global as well as local texture information. For example, a picture with an aircraft in mid air might be described as "sky, with a hole in it". To model this type of information, we use a proto-concept occurrence histogram where each bin is a proto-concept. The values in the histogram are the similarity responses of each proto-concept annotation, to the regions in the image.

We use the proto-concept occurrence histogram to characterize both global and local texture information. Global information is described by computing an occurrence histogram accumulated over all regions in the image. Local information is taken into account by constructing another occurrence histogram for only the response of the best region. For each proto-concept, or bin, b the accumulated occurrence histogram and the best occurrence histogram are constructed by,

$$\begin{aligned} H_{accumulated}(b) &= \sum_{r \in R(im)} \sum_{a \in A(b)} W2(a,r) \quad , \\ H_{best}(b) &= \arg\max_{r \in R(im)} \sum_{a \in A(b)} W2(a,r) \quad , \end{aligned}$$

where $R(im)$ denotes the set of regions in image im, A(b) represents the set of stored annotations for proto-concept b, and $W2$ is the Cramér-von Mises statistic as introduced in equation 2.

We denote a proto-concept occurrence histogram as a contexture for that image. We have chosen this name, as our method incorporates texture features in a context. The texture features are given by the use of Wiccest features, using color invariance and natural image statistics. Furthermore, context is taken into account by the combination of both local and global region combinations.

Contextures can be computed for different parameter settings. Specifically, we calculate the contextures at scales $\sigma = 1$ and $\sigma = 3$ of the Gaussian filter. Furthermore, we use two different region sizes, with ratios of $\frac{1}{2}$ and $\frac{1}{6}$ of the x-dimension and y-dimensions of the image. Moreover, contextures are based on one image, and not based on a shot. To generalize our approach to shot level, we extract 1 frame per second out of the video, and then aggregate the frames that belong to the same shot. We use two ways to aggregate frames: 1) average the contexture responses for all extracted frames in a shot and 2) keep the maximum response of all frames in a shot. This aggregation strategy accounts for information about the whole shot i, and information about accidental frames, which might occur with high camera motion. The combination of all these parameters yields a vector of contextures $\vec{v}_i$, containing the final result of the visual analysis.

3.1.3 Textual Analysis

In the textual modality, we aim to learn the association between uttered speech and semantic concepts. A detection system transcribes the speech into text. For the Chinese and Arabic sources we exploit the provided machine translations. The resulting translation is mapped from story level to shot level. From the text we remove the frequently occurring stopwords. After stopword removal, we are ready to learn semantics.

To learn the relation between uttered speech and concepts, we connect words to shots. We make this connection within the temporal boundaries of a shot. We derive a lexicon of uttered words that co-occur with ω using the shot-based annotations of the training data. For each concept ω, we learn a separate lexicon, Λ_T^ω, as this uttered word lexicon is specific for that concept. For feature extraction we compare the text associated with each shot with Λ_T^ω. This comparison yields a text vector $\vec{t}_i$ for shot i, which contains the histogram of the words in association with ω.

3.1.4 Early Fusion

Indexing approaches that rely on early fusion first extract unimodal features of each stream. The extracted features of all streams are combined into a single representation. After combination of unimodal features in a multimodal representation, early fusion methods rely on supervised learning to classify semantic concepts. Early fusion yields a truly multimedia feature representation, since the features are integrated from the start. An added advantage is the requirement of one learning phase only. Disadvantage of the approach is the difficulty to combine features into a common representation. The general scheme for early fusion is illustrated in Fig. 5a.

We rely on vector concatenation in the early fusion scheme to obtain a multimodal representation. We concatenate the visual vector $\vec{v}_i$ with the text vector $\vec{t}_i$. After feature normalization, we obtain early fusion vector $\vec{e}_i$.

3.1.5 Late Fusion

Indexing approaches that rely on late fusion also start with extraction of unimodal features. In contrast to early fusion, where features are then combined into a multimodal representation, approaches for late fusion learn semantic concepts directly from unimodal features. In general, late fusion schemes combine learned unimodal concept scores into a multimodal representation. Then late fusion methods rely on supervised learning to classify semantic concepts. Late fusion focuses on the individual strength of modalities. Unimodal concept detection scores are fused into a multimodal

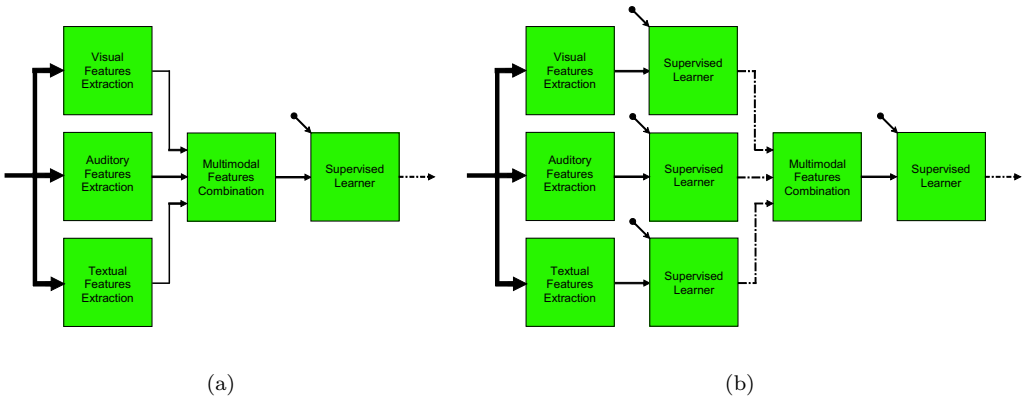

Figure 5: (a) General scheme for early fusion. Output of unimodal analysis is fused before a concept is learned. (b) General scheme for late fusion. Output of unimodal analysis is used to learn separate scores for a concept. After fusion a final score is learned for the concept. We use the conventions of Fig. 1.

semantic representation rather than a feature representation. A big disadvantage of late fusion schemes is its expensiveness in terms of the learning effort, as every modality requires a separate supervised learning stage. Moreover, the combined representation requires an additional learning stage. Another disadvantage of the late fusion approach is the potential loss of correlation in mixed feature space. A general scheme for late fusion is illustrated in Fig. 5b.

For the late fusion scheme, we concatenate the probabilistic output score after visual analysis, i.e. $p_i^*(\omega|\vec{v}_i, \vec{q}^*)$, with the probabilistic score resulting from textual analysis, i.e. $p_i^*(\omega|\vec{t}_i, \vec{q}^*)$, into late fusion vector $\vec{l}_i$.

3.1.6 Content Pathfinder

We learn 101 semantic concepts based on the four vectors resulting from analysis in the content analysis step. Thus $\vec{v}_i, \vec{t}_i, \vec{e}_i$, and $\vec{l}_i$ serve as the input for our supervised learning module, which learns an optimized SVM model for each semantic concept ω using 3-fold cross validation with 3 repetitions on training set A. These models are then validated on set D, yielding a best performing model $p_i^*(\omega|\vec{m}_i)$ for all ω in Λ_S, where $\vec{m}_i \in \{\vec{v}_i, \vec{t}_i, \vec{e}_i, \vec{l}_i\}$.

3.2 Style Analysis Step

In the style analysis step we conceive of a video from the production perspective. Based on the four roles involved in the video production process [31], this step analyzes a video by four related style detectors. Layout detectors analyze the role of the editor. Content detectors analyze the role of production design. Capture detectors analyze the role of the production recording unit. Finally, context detectors analyze the role of the preproduction team, see Fig. 6.

3.2.1 Style Analysis

We develop detectors for all four production roles as feature extraction in the style analysis step. We refer to our previous work for specific implementation details of the detectors [31, Electronic Appendix]. We have chosen to convert the output of all style detectors to an ordinal scale, as this allows for elegant fusion.

For the layout $\mathcal{L}$ the length of a camera shot is used as a feature, as this is known to be an informative descriptor for genre [31]. Overlayed text is another informative descriptor. Its presence is detected by a text localization

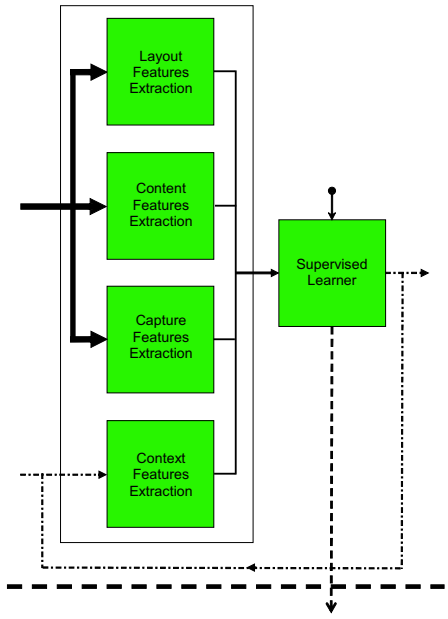

Figure 6: Feature extraction and classification in the style analysis step, special case of Fig. 2.

algorithm [25]. To segment the auditory layout, periods of speech and silence are detected based on the provided automatic speech recognition results. We obtain a voice-over detector by combining the speech segmentation with the camera shot segmentation [31]. The set of layout features is thus given by: $\mathcal{L} = \{$*shot length, overlayed text, silence, voice-over*$\}$.

As concerns the content $\mathcal{C}$, a frontal face detector [27] is applied to detect people. We count the number of faces, and for each face its location is derived [31]. In addition, we measure the average amount of object motion in a camera shot [29]. Based on provided speaker identification we identify each of the three most frequent speakers. Each camera shot is checked for presence of speech from one of the three [31]. We also exploit the provided named entity recognition. The set of content features is thus given by: $\mathcal{C} =\{$*faces, face location, object motion, frequent speaker, voice named entity*$\}$.

For capture $\mathcal{T}$, we compute the camera distance from the size of detected faces [27, 31]. It is undefined when no face is detected. In addition to camera distance, several types of camera work are detected [2], e.g. pan, tilt, zoom, and so on. Finally, for capture we also estimate the amount of camera motion [2]. The set of capture features is thus given by: $\mathcal{T} = \{$*camera distance, camera work, camera motion*$\}$.

The context $\mathcal{S}$ serves to enhance or reduce the correlation between semantic concepts. Detection of *vegetation* can aid in the detection of a *forest* for example. Likewise, the co-occurrence of a *space shuttle* and a *bicycle* in one shot is improbable. As the performance of semantic concept detectors is unknown and likely to vary between concepts, we exploit iteration to add them to the context. The rationale here is to add concepts that are relatively easy to detect first. They aid in detection performance by increasing the number of true positives or reducing the number of false positives. To prevent bias from domain knowledge, we use the performance on validation set D of all concepts from Λ_S in the content analysis step as the ordering for the context. To assign detection results for the first and least difficult concept, we rank all shot results on $p_i^*(\omega_1|\vec{m}_i)$. This ranking is then exploited to categorize results for ω_1 into one of five levels. The basic set of context features is thus given by: $\mathcal{S} = \{$*content analysis step* $\omega_1\}$.

The concatenation of $\{\mathcal{L}, \mathcal{C}, \mathcal{T}, \mathcal{S}\}$ for shot i yields style vector $\vec{s}_i$. This vector forms the input for an iterative classifier [31] that trains a style model for each concept in lexicon Λ_S. We classify all ω in Λ_S again in the style analysis step. We use 3-fold cross validation with 3 repetitions on training set B to optimize parameter settings in this analysis step. We use the resulting probability as output for concept detection in the style analysis step.

3.3 Context Analysis Step

The context analysis step adds context to our interpretation of the video. Our ultimate aim is the reconstruction of the author's intent by considering detected concepts in context.

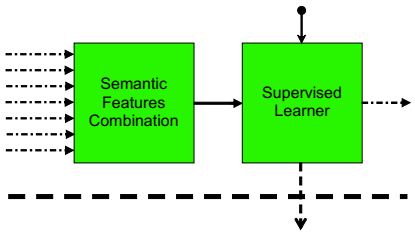

Figure 7: Feature extraction and classification in the context analysis step, special case of Fig. 2.

Both the content analysis step and the style analysis step yield a probability for each shot i and all concepts ω in Λ_S. The probability indicates whether a concept is present. We fuse these semantic features of an analysis step for a shot i into a context vector, see Fig. 7.

We consider three paths in the context analysis step. The first path stems directly from the content analysis step. We fuse the 101 $p_i^*(\omega|\vec{m}_i)$ concept scores into context vector $\vec{d}_i$. The second path stems from the style analysis step where we fuse the 101 $p_i^*(\omega|\vec{s}_i)$ scores into context vector $\vec{p}_i$. The third path selects the best performer on validation set D from either content analysis step or style analysis step. These best performers are fused in context vector $\vec{b}_i$.

From these three vectors we learn relations between concepts automatically. To that end the vectors serve as the input for a supervised learning module, which associates a contextual probability $p_i^*(\omega|\vec{c}_i)$ to a shot i for all ω in Λ_S, where $\vec{c}_i \in \{\vec{d}_i, \vec{p}_i, \vec{b}_i\}$. To optimize parameter settings, we use 3-fold cross validation with 3 repetitions on the previously unused data from training set C.

The output of the context analysis step is also the output of the entire semantic pathfinder on video documents. On the way we have included in the semantic pathfinder, the results of the analysis on raw data, facts derived from production by the use of style features, and a context perspective of the author's intent by using semantic features. For each concept we obtain several probabilities based on (partial) content, style, and context. We select from all possibilities the one that maximizes average precision based on performance on validation set D. The semantic pathfinder provides us with the opportunity to decide whether a one-shot analysis step is best for the concept only concentrating on (visual) content, or a two-analysis step classifier increasing discriminatory power by adding production style to content, or that a concept profits most from a consecutive analysis on content, style, and context level.

3.4 Experiments

We traversed the entire semantic pathfinder for all 101 concepts. The average precision performance of the semantic pathfinder and its sub-systems, on validation set D, are shown in Fig. 8.

We evaluated for each concept four analysis strategies in the content analysis step: text-only, visual-only, early fu-

Table 1: UvA-MediaMill TRECVID 2005 run comparison for all 10 benchmark concepts. The best path of the semantic pathfinder is marked in bold. Last column indicates results of our visual-only run.

	SP-1	**SP-2**	**SP-3**	**SP-4**	**SP-5**	**SP-6**	**Visual-only**
People walking	**0.199**	0.172	0.154	0.179	0.101	0.103	0.031
Explosion	**0.041**	0.027	0.032	0.035	0.036	0.034	0.073
Map	0.142	**0.16**	0.135	0.123	0.099	0.127	0.138
US flag	0.1	0.063	0.11	0.095	0.072	**0.114**	0.129
Building	**0.235**	0.229	0.226	0.225	0.21	0.157	0.269
Waterscape	**0.201**	0.198	0.137	0.164	0.124	0.136	0.166
Mountain	**0.22**	0.193	0.182	0.195	0.17	0.128	0.207
Prisoner	**0.005**	0.001	0	0.001	0.001	0.001	0.003
Sports	**0.342**	0.225	0.289	0.202	0.137	0.153	0.272
Car	**0.213**	0.192	0.182	0.201	0.196	0.199	0.233
MAP	**0.1698**	0.146	0.1447	0.142	0.1146	0.1152	0.1521

sion, and late fusion. Results confirm the importance of visual analysis for generic concept detection. Text-analysis yields the best approach for only 8 concepts, whereas visual analysis yields the best performance for as much as 45 concepts. Fusion is optimal for the remaining 48 concepts, with a clear advantage for early fusion (33 concepts) in favor of late fusion (15 concepts).

The style analysis step again confirms the importance for inclusion of professional television production facets for semantic video indexing. Especially for concepts which share many similarities in their production process, like anchors, monologues, and entertainment. For other concepts, content is more decisive, like tennis and baseball for example. Thus some concepts are just content, whereas others are pure production style.

We boost concept detection performance further by the usage of context. The pathfinder again exploits variation in performance for the various paths to select an optimal pathway. The results demonstrate the virtue of the semantic pathfinder. Concepts are divided by the analysis step after which they achieve best performance. Based on these results we conclude that an optimal strategy for generic multimedia analysis is one that learns from the training set on a per-concept basis which tactic to follow.

3.4.1 Pathfinder Runs

We submitted six paths for each benchmark concept, prioritized according to validation set performance. For concept *explosion* for example, the optimal path (SP-1) indicates that visual-only analysis is the best performer. However, in most cases the best path is a consecutive path of content, style, and context. We report the official TRECVID benchmark results in Table 1.

The results show that the pathfinder mechanism is a good way to estimate the best performing analysis path. The SP-1 run containing the optimal path is indeed the best performer in 8 out of 10 cases. Overall, this is also our best performing run. However, what strikes us most is that average precision results are much lower than can be expected based on validation set performance reported in Fig. 8. This may indicate that despite the use of separate training and validation sets we are still overfitting the data. A point of concern here is the random assignment of shots to the separate training and validation sets. This may bias the classifiers as it is possible that similar news items from several channels are distributed to separate sets. For two concepts (map and explosion) performance suffered from misinterpretation of correct concepts. Had we included examples of news anchors with maps in the background of the studio setting (for the map concept) and smoke (for explosion) in our training sets, results would be higher. When looking at the judged results, we also found that three concepts (waterscape, mountain, and car) are dominated by commercials. We do not perform well on commercial detection. This can be explained because we take 1 frame per second out of the video in the visual analysis. Sampling in this manner will select different frames for the same commercials that reappear on different time stamps in a video. We anticipate that improvement in frame sampling yields increased robustness for the entire pathfinder.

3.4.2 Visual-only Run

Validation set performance in Fig. 8. indicates that our visual analysis step performs quite good. To determine the contribution of the visual analysis step, we therefore submitted a visual-only run. This involved training a Support Vector Machine on the vector of contextures as introduced in section 3.1.1. We trained an SVM for each of the 10 concept of the concept detection task. An experiment for recognizing proto-concept was submitted by another group [37].

The visual features in the submitted visual-only run are slightly different from the visual features in the semantic pathfinder system. This difference is caused by ongoing development on the visual analysis. Specifically, we improved the Weibull fit to be more robust and we added the proto-concept *car*. The newer version of the visual analysis was not incorporated in the semantic pathfinder. It was not integrated because visual analysis is the first step in the semantic path. Thus, a change in the visual analysis means that all further paths would have to be recomputed. However, for a visual-only run, the improvements were feasible to compute.

Semantic Concept	Text Analysis	Visual Analysis	Early Fusion	Late Fusion	Style	Content-Context	Style-Context	Best-Context	Optimal Path
1 aircraft	0.049	0.199	0.203	0.157	0.093	0.205	0.110	0.210	0.210
2 allawi	0.188	0.054	0.229	0.026	0.011	0.274	0.007	0.243	0.274
3 anchor	0.175	0.585	0.472	0.562	0.764	0.615	0.780	0.771	0.780
4 animal	0.209	0.189	0.216	0.181	0.316	0.330	0.301	0.417	0.417
5 arrafat	0.084	0.112	0.073	0.078	0.135	0.141	0.247	0.176	0.247
6 baseball	0.051	0.240	0.226	0.040	0.085	0.084	0.073	0.028	0.240
7 basketball	0.033	0.541	0.235	0.451	0.532	0.573	0.589	0.641	0.641
8 beach	0.002	0.005	0.005	0.002	0.036	0.009	0.011	0.010	0.036
9 bicycle	0.096	0.025	0.128	0.098	0.140	0.109	0.406	0.400	0.406
10 bird	0.201	0.716	0.379	0.454	0.487	0.717	0.462	0.678	0.717
11 boat	0.065	0.147	0.039	0.169	0.102	0.172	0.132	0.222	0.222
12 **building**	0.159	0.281	0.251	0.085	0.292	0.298	0.304	0.327	0.327
13 bus	0.101	0.025	0.095	0.146	0.024	0.015	0.021	0.018	0.146
14 bush_jr	0.072	0.173	0.072	0.144	0.213	0.201	0.224	0.219	0.224
15 bush_sr	0.028	0.019	0.021	0.001	0.217	0.065	0.198	0.205	0.217
16 candle	0.008	0.003	0.020	0.024	0.006	0.002	0.003	0.018	0.024
17 **car**	0.108	0.253	0.197	0.214	0.215	0.269	0.243	0.282	0.282
18 cartoon	0.511	0.747	0.569	0.640	0.455	0.601	0.528	0.693	0.747
19 chair	0.100	0.534	0.328	0.522	0.207	0.552	0.284	0.577	0.577
20 charts	0.209	0.275	0.440	0.384	0.321	0.456	0.322	0.463	0.463
21 clinton	0.002	0.264	0.075		0.207	0.018		0.002	0.264
22 cloud	0.034	0.237	0.101	0.156	0.126	0.228	0.128	0.172	0.237
23 corporate_leader	0.040	0.097	0.051	0.077	0.078	0.049	0.080	0.065	0.097
24 court	0.077	0.057	0.338	0.003	0.099	0.350	0.116	0.368	0.368
25 crowd	0.233	0.404	0.404	0.402	0.391	0.424	0.414	0.446	0.446
26 cycling	0.103	0.020	0.135	0.001	0.435	0.121	0.428	0.421	0.435
27 desert	0.034	0.114	0.129	0.098	0.070	0.143	0.095	0.144	0.144
28 dog	0.284	0.262	0.446	0.004	0.294	0.483	0.200	0.498	0.498
29 drawing	0.318	0.275	0.269	0.318	0.045	0.208	0.029	0.274	0.318
30 drawing_cartoon	0.403	0.288	0.293	0.405	0.093	0.442	0.219	0.443	0.443
31 duo_anchor	0.008	0.651	0.054	0.060	0.857	0.602	0.881	0.882	0.882
32 entertainment	0.257	0.268	0.325	0.193	0.684	0.496	0.693	0.700	0.700
33 **explosion**	0.040	0.127	0.087	0.060	0.094	0.118	0.034	0.125	0.127
34 face	0.724	0.898	0.893	0.755	0.913	0.696	0.925	0.929	0.929
35 female	0.065	0.316	0.118	0.021	0.414	0.336	0.419	0.420	0.420
36 fireweapon	0.036	0.039	0.128	0.043	0.037	0.131	0.055	0.059	0.131
37 fish	0.065	0.235	0.116	0.100	0.284	0.231	0.322	0.353	0.353
38 flag	0.096	0.165	0.121	0.157	0.135	0.182	0.145	0.184	0.184
39 **flag_usa**	0.077	0.185	0.141	0.175	0.137	0.190	0.162	0.215	0.215
40 food	0.016	0.071	0.068	0.030	0.172	0.138	0.187	0.216	0.216
41 football	0.026	0.188	0.088	0.033	0.252	0.196	0.330	0.351	0.351
42 golf	0.069	0.038	0.179	0.092	0.109	0.190	0.059	0.214	0.214
43 government_building	0.026	0.035	0.019	0.157	0.212	0.008	0.212	0.213	0.213
44 government_leader	0.291	0.275	0.261	0.378	0.400	0.401	0.412	0.416	0.416
45 graphics	0.169	0.354	0.358	0.340	0.363	0.445	0.402	0.472	0.472
46 grass	0.016	0.151	0.042	0.063	0.098	0.167	0.094	0.107	0.167
47 hassan_nasrallah	0.446	0.867	0.278	0.667	0.158	0.917	0.251	1.000	1.000
48 horse	0.001	0.129	0.219	0.001	0.308	0.182	0.341	0.338	0.341
49 horse_racing	0.001	0.059	0.253	0.201	0.540	0.204	0.409	0.406	0.540
50 house	0.081	0.005	0.081	0.006	0.012	0.005	0.014	0.008	0.081
51 hu_jintao	0.267	0.094	0.230	0.082	0.060	0.296	0.069	0.323	0.323
52 indoor	0.400	0.616	0.584	0.607	0.677	0.674	0.718	0.722	0.722
53 kerry	0.030	0.079	0.028	0.005	0.028	0.123	0.003	0.065	0.123
54 lahoud	0.135	0.394	0.248	0.297	0.258	0.559	0.330	0.454	0.559
55 male	0.101	0.244	0.131	0.215	0.279	0.259	0.291	0.294	0.294
56 **maps**	0.146	0.406	0.308	0.323	0.388	0.471	0.407	0.493	0.493
57 meeting	0.202	0.368	0.228	0.352	0.393	0.404	0.422	0.452	0.452
58 military	0.183	0.239	0.305	0.331	0.282	0.357	0.293	0.358	0.358
59 monologue	0.053	0.128	0.089	0.138	0.692	0.149	0.718	0.724	0.724
60 motorbike	0.003	0.399	0.163	0.003	0.014	0.389			0.399
61 **mountain**	0.041	0.299	0.181	0.203	0.228	0.347	0.250	0.331	0.347
62 natural_disaster	0.126	0.035	0.152	0.106	0.056	0.151	0.028	0.163	0.163
63 newspaper	0.068	0.526	0.433	0.454	0.497	0.525	0.497	0.529	0.529
64 nightfire	0.011	0.009	0.009	0.003	0.005	0.131	0.002	0.003	0.131
65 office	0.029	0.073	0.065	0.091	0.071	0.062	0.078	0.098	0.098
66 outdoor	0.440	0.668	0.706	0.665	0.634	0.744	0.726	0.754	0.754
67 overlayed_text	0.552	0.697	0.678	0.686	0.991	0.706	0.991	0.990	0.991
68 people	0.803	0.833	0.870	0.804	0.937	0.848	0.890	0.926	0.937
69 people_marching	0.121	0.229	0.232	0.169	0.218	0.252	0.227	0.256	0.256
70 police_security	0.017	0.007	0.015	0.009	0.019	0.017	0.018	0.022	0.022
71 powell	0.033	0.019	0.073	0.012	0.019	0.031	0.190	0.077	0.190
72 **prisoner**	0.011	0.008	0.077	0.003	0.011	0.088	0.013	0.088	0.088
73 racing	0.007	0.009	0.006	0.001	0.008	0.010	0.029	0.051	0.051
74 religious_leader	0.268	0.060	0.251	0.190	0.022	0.252	0.006	0.346	0.346
75 river	0.167	0.500	0.084	0.252	0.017	0.025	0.061	0.120	0.500
76 road	0.120	0.239	0.219	0.219	0.230	0.268	0.252	0.277	0.277
77 screen	0.110	0.066	0.126	0.075	0.073	0.154	0.080	0.149	0.154
78 sharon	0.003	0.008	0.210	0.037	0.008	0.199	0.002	0.151	0.210
79 sky	0.180	0.499	0.498	0.494	0.482	0.537	0.497	0.551	0.551
80 smoke	0.084	0.330	0.272	0.282	0.219	0.374	0.208	0.353	0.374
81 snow	0.066	0.036	0.101	0.028	0.084	0.299	0.142	0.056	0.299
82 soccer	0.037	0.533	0.365	0.455	0.510	0.578	0.512	0.636	0.636
83 splitscreen	0.080	0.616	0.287	0.591	0.819	0.677	0.757	0.795	0.819
84 **sports**	0.132	0.296	0.257	0.320	0.423	0.459	0.466	0.529	0.529
85 studio	0.412	0.653	0.630	0.674	0.746	0.718	0.780	0.781	0.781
86 swimmingpool	0.002	0.001	0.001		0.178	0.012	0.181	0.175	0.181
87 table	0.083	0.135	0.140	0.083	0.203	0.107	0.176	0.197	0.203
88 tank	0.012	0.024	0.030	0.019	0.001	0.335	0.001	0.001	0.335
89 tennis	0.219	0.644	0.617	0.691	0.382	0.763	0.420	0.764	0.764
90 tony_blair	0.750	0.254	0.688	0.256	0.005	0.059	0.021	0.751	0.751
91 tower	0.015	0.023	0.083	0.020	0.068	0.062	0.073	0.115	0.115
92 tree	0.013	0.178	0.187	0.110	0.097	0.189	0.145	0.151	0.189
93 truck	0.040	0.035	0.049	0.022	0.051	0.062	0.066	0.068	0.068
94 urban	0.205	0.270	0.291	0.297	0.285	0.320	0.331	0.356	0.356
95 vegetation	0.071	0.224	0.198	0.188	0.204	0.236	0.210	0.240	0.240
96 vehicle	0.135	0.281	0.273	0.278	0.286	0.326	0.315	0.343	0.343
97 violence	0.233	0.291	0.338	0.348	0.387	0.451	0.440	0.485	0.485
98 **walking_running**	0.224	0.327	0.328	0.354	0.414	0.410	0.421	0.464	0.464
99 **waterbody**	0.077	0.275	0.203	0.237	0.251	0.305	0.289	0.346	0.346
100 waterfall	0.001	0.001	0.008		0.118	0.009	0.042	0.256	0.256
101 weather	0.461	0.240	0.508	0.483	0.555	0.579	0.560	0.548	0.579
MAP	**0.143**	**0.254**	**0.231**	**0.224**	**0.263**	**0.300**	**0.282**	**0.352**	**0.382**
TRECVID MAP	**0.101**	**0.246**	**0.203**	**0.197**	**0.245**	**0.296**	**0.259**	**0.320**	**0.322**

Figure 8: Validation set average precision performance for 101 semantic concepts using sub-systems of the semantic pathfinder. The best path for each concept is marked with gray cells. Empty cells indicate impossibility to learn models, due to lack of annotated examples in the training sub-set used.

Table 2: Validation set average precision performance for 3 types of camera work using several versions of our camera work detector.

	Pan	Tilt	Zoom	MAP
Late Fusion	0.862	0.786	0.862	0.837
Late Fusion + Selected Context	0.859	0.752	0.866	0.826
Late Fusion + Context	0.856	0.656	0.856	0.789
Early Fusion	0.703	0.558	0.783	0.681
Global	0.569	0.613	0.813	0.665
Global + Context	0.591	0.562	0.792	0.648
Early Fusion + Context	0.616	0.461	0.765	0.614

The results of our visual-run reflect the importance of visual analysis. For four concepts (explosion, US flag, building, car) we outperform the pathfinder system. This improvement might be attributed to the use of improved visual features and to the fact that we use the entire training set in SVM-training. However, since the visual analysis step is embedded in the pathfinder system, the visual analysis should never perform better. Therefore we believe that results of the pathfinder system will improve when the new features are included.

4 Camera Work

For the detection of camera work we start with an existing implementation based on spatiotemporal image analysis [34, 12]. Given a set of global intensity images from shot i, the algorithm first extract spatiotemporal images. On these images a direction analysis is applied to estimate direction parameters. These parameters form the input for a supervised learning module to learn three types of camera work. We modified the algorithm in various ways. We superimposed a tessellation of 8 regions on each input frame to decrease the effect of local disturbances. Parameters thus obtained are exploited using an early fusion and late fusion approach. In addition we explored whether the 101 concept scores obtained from the semantic pathfinder aid in detection of camera work.

4.1 Experiments

Experiments on validation set D indicate that average precision results increase drastically, especially for pan (+51%) and tilt (+28%), see Table 2. The best approach is a late fusion scheme without the usage of context. Relative to other participants we performed quite good in precision, but quite bad in terms of recall. Results indicate that the base detector is too conservative. However, it also shows that any global image based camera work detector has the potential to profit from a tessellation of region-based detectors.

5 Lexicon-driven Retrieval

We propose a lexicon-driven retrieval paradigm to equip users with semantic access to multimedia archives. The aim is to retrieve from a multimedia archive S, which is composed of n unique shots $\{s_1, s_2, \ldots, s_n\}$, the best possible answer set in response to a user information need. To that end, we use the 101 concepts in the lexicon as well as the 3 types of camera work for our automatic, manual, and interactive search systems.

5.1 Automatic Search

Our automatic search engine uses only topic text as input [10], as we postulate that it is unreasonable to expect a user to provide a video search system with example videos in a real world scenario. We rely purely on text and the lexicon of 101 semantic concept detectors that we have developed using the semantic pathfinder, see Section 3, to search through the video collection. We developed our search system using the video data, topics, and ground truths from the 2003 and 2004 TRECVID evaluations as a training set.

5.1.1 Indexing Components

Our automatic search system incorporates regular TFIDF-based indices for standard retrieval using the bfx-bfx [24] formula, Latent Semantic Indexing [5] for text retrieval with implicit query expansion, and 101 the different semantic concept indices for query-by-concept. Each index was matched to one or more concepts, or *synsets* in the WordNet [13] lexical database on an individual basis, according to whether the concept directly matches the content of the detectors. For example, the detector for the concept *baseball* finds shots of baseball games, and these shots invariably include baseball players, baseball equipment, and a baseball diamond, so these concepts are also matched. Additional synsets are added to WordNet for semantic concepts that do not have a direct WordNet equivalent.

5.1.2 Automatic Query Interface Selection

We perform the standard stopping and stemming procedures on the topic text (using the SMART stop list [23] with the addition of the words *find* and *shots*; and the Porter stemming algorithm [20] respectively). In addition, we perform part-of-speech tagging and chunking using the TreeTagger [26]. This grammatical information is used to identify two different query categorizations: complex vs. simple queries and general vs. specific queries. Any topic containing more than one noun chunk is classified as complex, as it refers to more than one object, while requests containing only a single noun chunk are classified as simple. If a request contains a name (a proper noun) it refers to a specific object, rather than a general category, so we categorize all requests containing proper nouns as specific requests, and all others as general requests.

Subsequently, we extract the WordNet words in the topic text through dictionary lookup of noun chunks and nouns. We identify the correct synset for WordNet words with multiple meanings through disambiguation. We evaluated

a number of disambiguation strategies using the WordNet::Similarity [17] resource, and found that for the purposes of our system, the best approach was to choose the most commonly occurring meaning of a word. Then we look for related semantic concept index synsets in the hypernym and hyponym trees of each of the topic synsets. If an index synset is found, we calculate the similarity between the two synsets using the Resnik similarity measure [21].

Finally, queries are formed. We create both a stemmed and an unstemmed TFIDF query using all of the topic terms. We create an extra TFIDF query on proper nouns only for specific topics, and a query on all nouns only for general topics. For the LSI index we create also a query using all of the topic terms, and in addition we create an additional query using proper nouns only for specific topics, and all nouns for general topics. Finally, we select the concept index with the highest Resnik similarity to a topic synset as the best match, and query on this concept.

5.1.3 Combining Query Results

We use a tiered approach for result fusion, first fusing the text results from the TFIDF and LSI searches individually, then fusing the resultant two sets, and finally combining them with the results from the semantic concept search. We use weighted Borda fusion to combine results, and developed the weights through optimization experiments on the training set. We use results from unstemmed searches to boost stemmed results for simple topics, as these benefit from using the exact spelling to search on text. We also boost text searches with a search on proper nouns for specific topics, as proper nouns are a good indicator of result relevance.

When combining text results with concept results, we use two measures developed specifically for WordNet by Resnik [21]: concept information content and similarity (previously mentioned). The information content measure is a measure of the specificity of a concept – as a concept becomes more abstract, the information content decreases. When the matching index concept has high information content, and the words in the concept do not, we give priority to the concept results. Likewise, when the matched concept index is very similar to the topic, then we give the concept search a very high weighting.

5.2 Manual Search

Our manual search approach investigates the power of lexicon driven retrieval used in a visual-only setting. We put the principle of lexicon driven retrieval to the test by using only the 101 concepts in answering the queries. Furthermore, we test the hypothesis that visual information, this year, is significantly more important than textual information. To test the impact of visual information, we use no other modality whatsoever, and rely only on visual features. This entails training a Support Vector Machine on the vector of contextures as introduced in section 3.1.1. This SVM is trained for every one of the 101 concepts with the whole development set as a training set. This lexicon of 101 visual concepts is subsequently used in answering the queries. For each query, we manually select one or two concepts that fit the question, and use the outcome of these detectors as our final answer to the question.

5.3 Interactive Search

Our interactive search systems stores the probabilities of all detected concepts and types of camera work for each shot in a database. In addition to learning, the paradigm also facilitates multimedia analysis at a similarity level. In the similarity component, 2 similarity functions are applied to index the data in the visual and textual modality. It results in 2 similarity distances for all shots, which are stored in a database. The MediaMill search engine offers users an access to the stored indexes and the video data in the form of 106 query interfaces; i.e. 2 query-by-similarity interfaces, 101 query-by-concept interfaces and 3 query-by-camera work interfaces. The query interfaces emphasize the lexicon-driven nature of the paradigm. Each query interface acts as a ranking operator Φ_i on the multimedia archive S, where $i \in \{1, 2, \ldots, 106\}$. The search engine stores results of each ranking operator in a ranked list ρ_i, which we denote by:

$$\rho_i = \Phi_i(S) \ . \tag{3}$$

The search engine handles the query requests, combines the results, and displays them to an interacting user. Within the paradigm, we perceive of interaction as a combination of querying the search engine and selecting relevant results using one of many display visualizations. A schematic overview of the retrieval paradigm is given in Fig. 9.

To support browsing with advanced visualizations the data is further processed. The high-dimensional feature space is projected to the 2D visualization space to allow for visual browsing. Clusters, and representatives for each cluster, are identified to support hierarchical browsing. Finally, semantic threads are identified, to allow for fast semantic browsing. For interactive search, users map topics to query-by-multimodal-concept or query-by-keyword to create a set of candidate results to explore. When there is a one-to-one relation between the query and the concept, a rank-time browsing method is employed. In other cases, the set forms the starting point for visual, hierarchical, or semantic browsing. The browsing methods are supported by advanced visualization and active learning tools.

5.3.1 Multimedia Similarity Indexing

After all the concepts are detected, the low level features are usually ignored. We believe, however, that these features are still valuable in adding information to the results of query-by-concept search. Except for specific concepts such as person X (*Allawi*, *Bush*, *Blair*), *USA flag*, most of provided concepts have general meaning like *sport*, *animal*, *maps*, *drawing*. These concepts can be classified further into

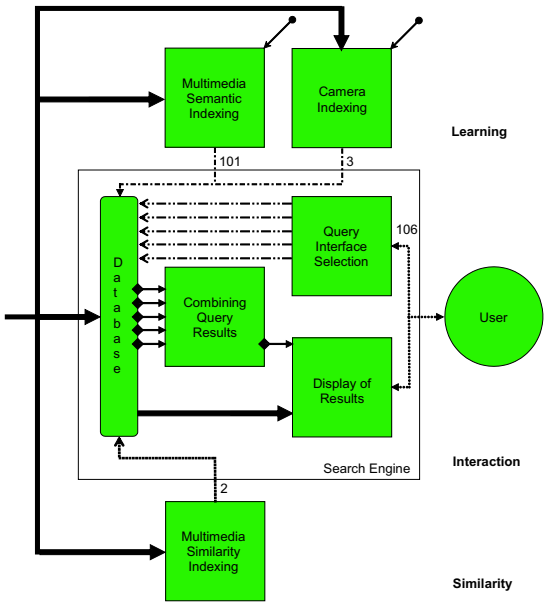

Figure 9: The lexicon-driven paradigm for interactive multimedia retrieval combines learning, similarity, and interaction. It learns to detect a lexicon of 101 semantic concepts together with 3 types of camera work. In addition, it computes 2 similarity distances. A search engine then presents 2 interfaces for query-by-similarity, 3 interfaces for query-by-camera-work, and 101 interfaces for query-by-concept. Based on interaction a user may refine search results until an acceptable standard is reached.

sub-concepts. For instance, the map concepts may contain maps in weather forecast, or a map of a country in a news report. Hence, we allow users to distinguish query-by-concept results further based on low level features.

There are different options for selecting low-level features, either using colors, textures, shapes or combinations of those. We use the visual concept features from the visual analysis step of the semantic pathfinder, see Section 3.1.1. We exploit the same 15 proto-concepts, but now with 6 different parameter sets for each shot. Those values are represented as a feature vector per shot. All the shots with their corresponding feature vectors built up a 90 dimensional feature space.

Obtaining the best performance on retrieving images, not only depends on the features, but also on the selection of an appropriate similarity function. The aim is to choose the best distance function that is able to return the maximum number of relevant images in its nearest neighbors. Based on experimental results we choose the L_2 measure as a distance function.

5.3.2 Combining Query Results

Combination by Linear Weighting To reorder ranked lists of results, we first determine the rank r_{ij} of shot s_j over the various ρ_i. Denoted by:

$$r_{ij} = \rho_i(s_j) \ . \tag{4}$$

We define a weight function $w(\cdot)$ that computes the weight of s_j in ρ_i based on r_{ij}. This linear weight function gives a higher weight to shots that are retrieved in the top of ρ_i and gradually reduces to 0. This function is defined as:

$$w(r_{ij}) = \frac{n - r_{ij} + 1}{n} \ . \tag{5}$$

We aggregate the results for each shot s_j by adding the contribution from each ranked list ρ_i. We then use the final ranking operator Φ^* to rank all shots from S in descending order based on this new weight. This combination method yields a final ranked list of results ρ^*, defined as:

$$\rho^* = \Phi^* \left(\left\{ \sum_i^m w(r_{ij}) \right\}_{j=1,2,\ldots,n} \right) \ , \tag{6}$$

where m indicates the number of selected query interfaces.

Combination by Semantic Threads The generated concept probabilities more or less describe the content of each shot. However, since there are only a limited number of categories for detection, a problem arises when a shot doesn't fit into any category, i.e. each individual concept detector returned a near-zero value. All shots with all concept values below a threshold could simply be removed. However some detectors produce low-value results but the top-ranked shots are still correct. This needs to be taken into account when combining shots. We use a round-robin pruning procedure to ensure that at least a top-N shots from each concept detector is included, even when that detector has very low values compared to other detectors.

Each remaining shot now contains at least one detected concept. With this information a distance measurement between shots can be created. But how do we measure distance between concept vectors? If we assume equal distances between concepts, we can construct a distance matrix made up from the similarity S_{pq} between shots p and q using well-known distance metrics such as Euclidean distance or histogram intersection. Given the computed distance between shots, it is possible to find groups of related shots using clustering techniques. Currently we use K-means clustering.

Now that clusters of related shots exist the task of forming a single coherent line of shots from each cluster must be examined. We apply a shortest path algorithm so that shots that are next to each other usually have a very low distance to each other, which means that shots with similar semantic content are near each other.

5.4 Display of Results

For effective interaction an interface for communicating between the user and the system is needed. We consider two issues that are required for an effective interface:

(1) For query specification, support should be given to explore the collection in search of good examples as the user seldom has a good example at his/her disposal.

Figure 10: Interfaces of the MediaMill semantic video search engine. On the left the CrossBrowser showing results for tennis. On top the SphereBrowser, displaying several semantic threads. Bottom right: active learning using a semantic cluster-based visualization in the GalaxyBrowser.

Most existing systems browse key frames in sequence (left-right, top-down) [28]. Hence, relations between frames are not taken into account. For effective interaction this may be unappropriate as the user can not benefit from the inherent structure found in video collections. Therefore,

(2) In the visualization, relations between key frames should be taken into account to allow selection of several frames by one user action.

For these reasons, visualization of key frames including support for browsing and exploring is essential in an interactive search system. We explored three advanced visualizations.

5.4.1 CrossBrowser

To visualize query-by-concept results we propose a *CrossBrowser*. The browser displays two orthogonal dimensions. The horizontal one is the time-thread, using the original TRECVID shot sequence. The vertical dimension contains the ranked list of query results. The GUI gives the user a cross layout of nearby shots on the screen. It exploits the observation that semantically similar shots tend to cluster in the time dimension. The resulting browser is visible in Fig. 10.

5.4.2 GalaxyBrowser

To speed up the search within the time limitation, we want to support the user with a system that they are able to select more than one key frame in one mouse action. It can be assumed that the key frames relevant to a search topic share similar features. Hence, they should be close to each other in the feature space. Therefore, visualization based on the similarity between them will make the search easier as similar images are grouped together in a specific location of the search space. Hence, less navigation and interaction actions will be needed. We propose the *GalaxyBrowser*, which integrates advanced similarity based visualization with active learning.

The similarity based visualization of [15] is the basis for our retrieval. In brief, we have pointed out that for an optimal visualization system, three requirements have to obeyed: *overview, structure preservation* and *visibility*. The first requirement ensures that the set displayed will be able to represent the whole collection, the so called representative set. For user interaction, the collection should be projected to the display space. Hence, the second requirement tries to preserve the relations between key frames in the original feature space. The final requirement keeps the content of displayed key frames feasible for interaction.

These are conflicting requirements. For example, to satisfy the overview requirement, the number of representative key frames should be increased. Because of the fixed size of the display space, the more key frames the higher the chance of overlap, the visibility requirement hence will be violated. Moreover, while preserving the visibility images are spread out from each other, original relations between them are changed i.e. structure is not preserved. Therefore, cost functions for each requirement and balancing functions between them are proposed.

Active learning algorithms mostly use support vector machines (SVM) as a feedback learning base [38, 33]. In interactive search, using this approach, the system first shows some images and asks the user to label those as positive and/or negative. The learning is either based on both positive and negative examples (known as two-class SVM) or on positive/negative ones only (known as one-class SVM). These examples are used to train the SVM to learn classifiers separating positive and negative examples. The process is repeated until the performance satisfies given constraints. We have done a comparison between the two approaches, the results turn out that one-class SVM generally performs better than the two-class, as well as faster in returning the result. We concentrate on the use of one-class SVM for learning the relevance feedback.

The combination of the two techniques is drawn into one scheme (see Fig. 11). The offline stage contains feature extraction and similarity function selection. The ISOSNE from [15] is applied to project the collection from the high dimensional space to the visualization space. The next step will decide which set of key frames will be used as a representative one. To do so, we employ k-means algorithm to cluster key frames into a fixed number of groups. A set of key frames selected from different groups is the representative set of the collection. Information of each key frame belonging to a certain group, and its position in the visualization space is stored as offline data.

In the interactive stage, query results are input for starting up the search. First, the set of top k key frames from the query results is displayed. The user then uses the system to explore the collection and find relevant key frames. Particularly, if the currently displayed set contains any positive one, the user selects that key frame and goes into the corresponding cluster with the expectation of finding more similar ones. With the advantage of similarity based visualization, instead of clicking on an individual key frame for labeling, the system supports the user with mouse dragging to draw the area of key frames in the same category. This means that when the user finds a group of relevant key frames, he/she draws a rectangle around those and marks them all as positive examples. Therefore, our system can reduce the number of actions from the user with the same amount of information for relevance feedback. In case there is no positive key frame in the current set, the user then asks the system to display another set, which contains the next k key frames from the query results. Key frames which are selected as training examples or displayed before will not be shown again.

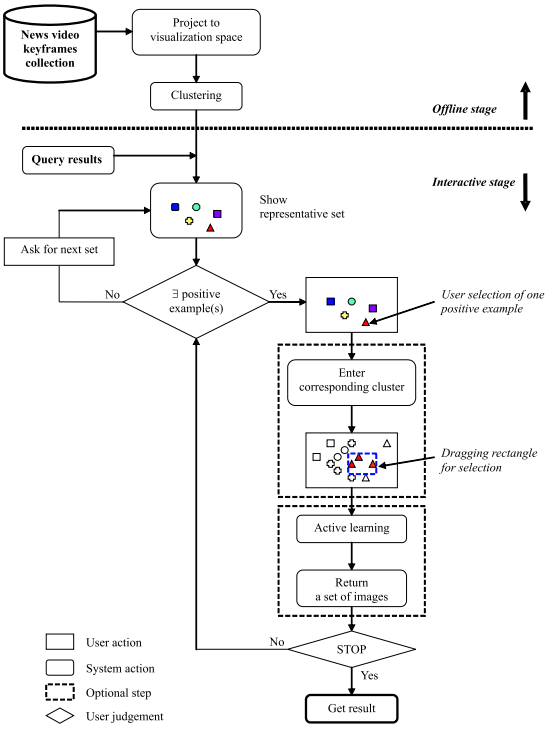

Figure 11: Scheme of an interactive search in the GalaxyBrowser with the combination of active learning and similarity based visualization.

In the learning step, when a certain number of training examples are provided, the SVM trains the support vectors. We use the well-known SVM library developed by Chang and Lin [4], which provides a one-class implementation. After the learning, a set of images closest to the border is returned. The process is repeated until a certain constraint is satisfied such as number of iterations, time limitation, or simply that the user does not want to give any more feedback. At that point, the system will return the final result containing key frames with maximum distances to the border as they are assumed having high probabilities to be relevant to the search topic.

5.4.3 SphereBrowser

To visualize the thread structure a so called SphereBrowser [22] was developed. The browser displays two orthogonal dimensions. The horizontal one is the time-thread, using the original TRECVID shot sequence. The vertical dimension contains for each shot cluster-threads of semantically similar footage. The GUI gives the user a a spherical layout of nearby shots on the screen, and the user can jump to any shown shot with transition animations between movements so that the browser gives the user the feeling he is looking at one side of a giant turnable sphere of video material. Using the mouse and arrow keys the user can then navigate either through time or through related shots, selecting relevant shots when found. Also selecting (parts

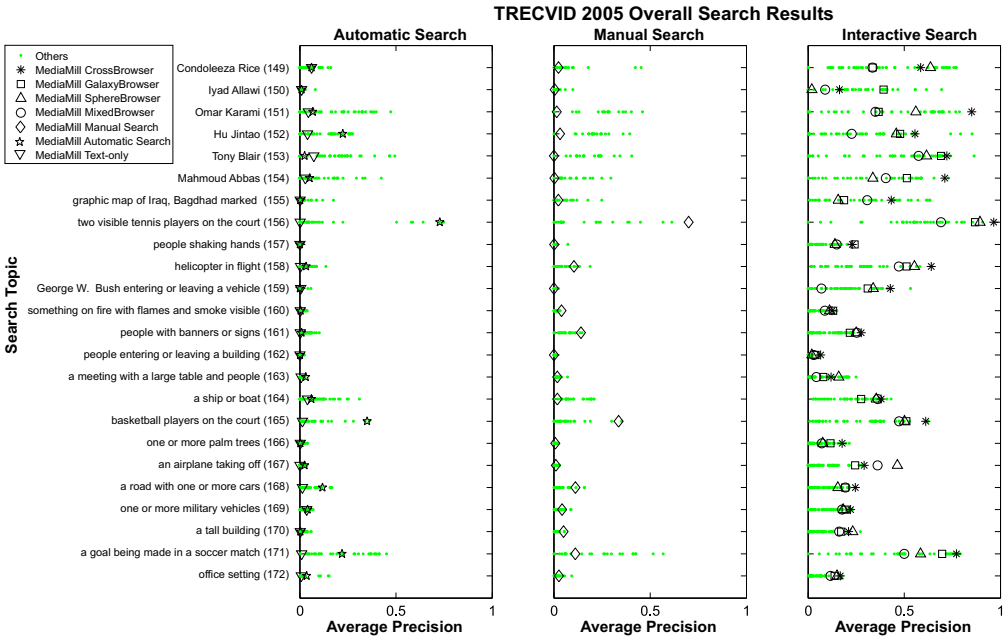

Figure 12: Comparison of automatic, manual, and interactive search results for 24 topics. Results for the users of the lexicon-driven retrieval paradigm are indicated with special markers.

of) entire threads is possible. Smooth transition animations exist to enable the user to have a better intuitive feeling of where he is browsing in the data set. The resulting browser is shown in Fig. 10.

5.5 Experiments

5.5.1 Automatic Search

We submitted two runs for automatic search, one baseline run using the final text search strategy only, and one full run incorporating text and semantic concepts. As can be seen in Fig. 12 the combined semantic and text run outperformed the text run on nearly all counts. We did best for those topics that had a clear mapping to the semantic concept indices, i.e. *tennis* for topic 156, *meeting* for topic 163 (achieving the best result for this topic) and *basketball* for topic 165. In some cases the concept weighting strategy was not optimal, for example for topic 158. In this case we detected the *aircraft* index, but the concept results were given a weighting of 0 in the result fusion because the information content of the concept *helicopter* was calculated to be much higher than the information content of the concept *aircraft*. If we had utilized the aircraft detector in this case, we would have achieved an average precision of 0.17, which is higher than the best evaluated average precision of 0.14.

We have demonstrated that automatic search using only text as input is a realistic task. We perform better than the median for a number of topics, and even achieve the best score for one topic. Postulating that all other systems incorporate multimodal examples in their search, this is a significant result. The performance of our search engine is best when one or more related indices are present; we expect that the results of our system will improve as we add more semantic concept indices, using our semantic pathfinder strategy.

5.5.2 Manual Search

We submitted one run for manual search where we only use the 101 concepts in the lexicon to answer the queries. Moreover, we restrict ourselves to using only visual information. For thirteen topics we score above the median. Specifically, for two queries, i.e. *vehicle with flames* (160) and *tennis players* (156) we perform the best of all manual runs, and for two other queries, i.e. *people with banners* (161) and *basketball players* (165) we are second best. For ten queries we score below the median, three of those are not covered by our lexicon, and seven are *person-x* type queries. We perform badly for *person-x* queries because the features describe visual scene layout, consequently, names and faces are not modeled. For the remaining fourteen topics there is only one i.e. *boat* (164) where we score below the median. Compared to our automatic search text baseline, we perform worse on eight queries. Of those eight queries, the text baseline performs better for all *person-x* queries, and for one other query (164). Consequently, a visual-only approach outperforms the text baseline in 16 queries, including the out-of-lexicon queries.

We believe our results support the lexicon-driven retrieval approach and show the importance of visual analysis. Despite the obvious disadvantages of using only visual information, we outperform the text baseline, and even score the best of all manual runs in two queries.

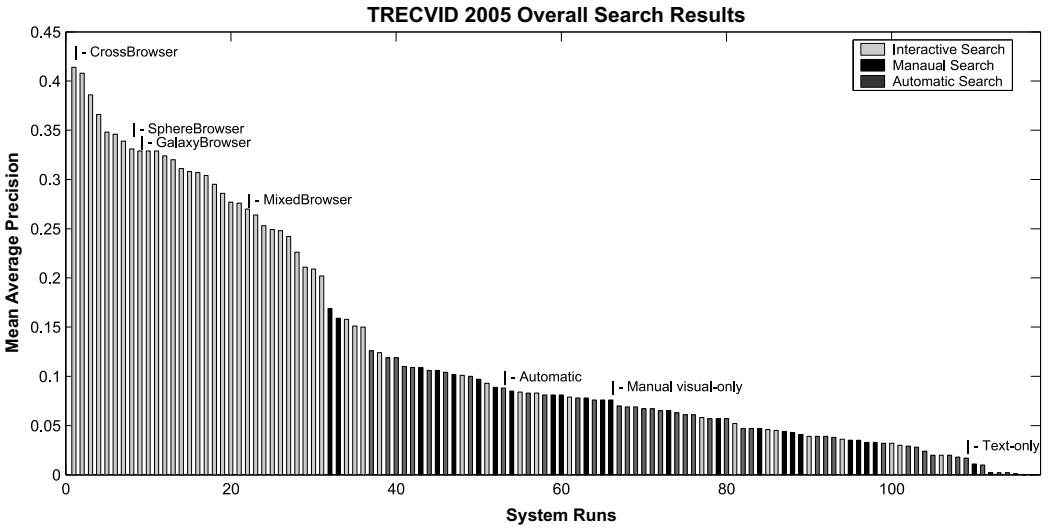

Figure 13: Overview of all search runs submitted to TRECVID 2005, ranked according to mean average precision. Users who exploited the proposed paradigm are indicated with special markers.

5.5.3 Interactive Search

We submitted four runs for interactive search. Three users focussed on using only one browser. The fourth users mixed all browsers. Results in Fig. 12 indicate that for most search topics, users of the proposed paradigm for interactive multimedia retrieval score above average. Furthermore, users of our approach obtain a top-3 average precision result for 19 out of 24 topics. Best performance is obtained for 7 topics. Best results are obtained with the CrossBrowser.

Depending on the search topic, the proposed GalaxyBrowser aids users in searching for the relevant subset of the collection. As the features used are visual based, the system works well in case relevant images of a certain topic share visual similarity, e.g. queries related to *tennis* or *car*. However, when topics have large variety in visual settings, for instance *person x* topics, visual features hardly yield additional information to aid the user in the interactive search process. To our knowledge, no existing features work well in these cases.

Two search strategies were discovered during the interactive retrieval task using the SphereBrowser. There were topics for which multiple cluster threads yielded good results for that topic, such as *Tennis* (156), *People with banners or signs* (161), *Meeting* (163) and *Tall building* (170). For these topics only the relevant parts of the threads needed to be selected. Another selection method was found in queries such as *Airplane takeoff* (167) and *Office setting* (172). Here there were only a limited number of consecutive valid shots visible in each thread, but because of the combination of both time and cluster threads there was always another valid but not yet selected shot visible. For these queries, selection was done by hopping from one valid result to another. Also a number of topics were not answerable by the SphereBrowser because of lack of nearby shots. These include *person x* topics 149, 151, and 153.

To gain insight in the overall quality of our lexicon-driven retrieval paradigm. We compare the results of our users with all other users that participated in the retrieval tasks of the 2005 TRECVID benchmark. We visualized the results for all submitted search runs in Fig. 13. The results are state-of-the-art.

6 Exploration of BBC Rushes

The BBC Rushes consist of raw material used to produce a video. Since there is little to no speech, this material is very suitable for visual-only indexing. We first segmented the video's using our shot segmentation algorithm [2]. Then we applied our best performing camera motion detector (see Section 4) on the BBC rushes using the models trained for the news data. To further investigate the robustness of our visual features, we performed visual-only concept detection on the BBC rushes data, without re-training the visual models. The visual models are the same as used in the visual only feature task (Section 3.4) and in the manual search task (Section 5.2). The detectors thus learned on news data are subsequently evaluated on the BBC rushes videos. Obviously, not all 101 concepts are useful, since they are trained on broadcast news. However, 25 concepts transcend the news domain and some perform surprisingly well on the BBC rushes: aircraft, bird, boat, building, car, charts, cloud, crowd, face, female, food, government building, grass, meeting, mountain, outdoor, overlayed text, sky, smoke, tower, tree, urban, vegetation, vehicle, water body. We developed a version of the MediaMill semantic video search engine tailored to the BBC rushers based on the computed indexes. While still primitive in terms of utility, the search engine allows users to explore the collection in a surprising manner. The results again confirm the importance of robust visual features. Hence, for this task much is to

be expected from improved visual analysis yielding a large lexicon of semantic concepts.

Acknowledgements

NIST for the evaluation effort. Kevin Walker for solving hard disk problems. DCU team for creation of key frames. Alex Hauptmann for missing machine translations.

References

[1] A. Amir et al. IBM research TRECVID-2003 video retrieval system. In *Proc. of the TRECVID Workshop*, Gaithersburg, USA, 2003.

[2] J. Baan et al. Lazy users and automatic video retrieval tools in (the) lowlands. In E. Voorhees and D. Harman, editors, *Proc. of the 10th Text REtrieval Conference*, volume 500-250, Gaithersburg, USA, 2001.

[3] Blinkx Video Search, 2005. `http://www.blinkx.tv/`.

[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] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. American Society Inform. Science*, 41(6):391–407, 1990.

[6] J. Geusebroek. Visual object recognition, 2005. Patent PCT/NL2005/000485 filed July 6.

[7] J. Geusebroek, R. Boomgaard, A. Smeulders, and H. Geerts. Color invariance. *IEEE Trans. PAMI*, 23(12):1338–1350, 2001.

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

[9] Google Video Search, 2005. `http://video.google.com/`.

[10] B. Huurnink. AutoSeek: Towards a fully automated video search system. Master's thesis, Universiteit van Amsterdam, 2005.

[11] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. PAMI*, 22(1):4–37, 2000.

[12] P. Joly and H.-K. Kim. Efficient automatic analysis of camera work and microsegmentation of video using spatiotemporal images. *Signal Processing: Image Communication*, 8(4):295–307, 1996.

[13] G. Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[14] M. Naphade. On supervision and statistical learning for semantic multimedia analysis. *J. Visual Commun. Image Representation*, 15(3):348–369, 2004.

[15] G. Nguyen and M. Worring. Similarity based visualization of image collections. In *Int'l Worksh. Audio-Visual Content and Information Visualization in Digital Libraries*, 2005.

[16] NIST. TRECVID Video Retrieval Evaluation, 2001–2005. `http://www-nlpir.nist.gov/projects/trecvid/`.

[17] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet:Similarity - measuring the relatedness of concepts. In *Nat'l Conf. Artificial Intelligence*, San Jose, USA, 2004.

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

[19] J. Platt. Probabilities for SV machines. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

[20] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[21] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *International Joint Conference on Artificial Intelligence*, San Mateo, USA, 1995.

[22] O. Rooij. Browsing news video using semantic threads. Master's thesis, Universiteit van Amsterdam, 2006.

[23] G. Salton. *The SMART retrieval system: experiments in automatic document processing*. Prentice-Hall, Englewood Cliffs, USA, 1971.

[24] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[25] T. Sato, T. Kanade, E. Hughes, M. Smith, and S. Satoh. Video OCR: Indexing digital news libraries by recognition of superimposed caption. *Multimedia Systems*, 7(5):385–395, 1999.

[26] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[27] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, 2004.

[28] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content based image retrieval at the end of the early years. *IEEE Trans. PAMI*, 22(12):1349–1380, 2000.

[29] C. Snoek and M. Worring. Multimedia event-based video indexing using time intervals. *IEEE Trans. Multimedia*, 7(4):638–647, 2005.

[30] C. Snoek, M. Worring, J. Geusebroek, D. Koelma, F. Seinstra, and A. Smeulders. The semantic pathfinder: Using an authoring metaphor for generic multimedia indexing. *IEEE Trans. PAMI*, 2006. In press.

[31] C. Snoek, M. Worring, and A. Hauptmann. Learning rich semantics from news video archives by style analysis. *ACM Trans. Multimedia Computing, Communications and Applications*, 2(2), May 2006. in press.

[32] C. Snoek, M. Worring, and A. Smeulders. Early versus late fusion in semantic video analysis. In *ACM Multimedia*, Singapore, 2005.

[33] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *ACM Multimedia*, pages 107–118, Ottawa, Canada, 2001.

[34] Y. Tonomura, A. Akutsu, Y. Taniguchi, and G. Suzuki. Structured video computing. *IEEE Multimedia*, 1(3):34–43, 1994.

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

[36] T. Volkmer, J. Smith, A. Natsev, M. Campbell, and M. Naphade. A web-based system for collaborative annotation of large image and video collections. In *ACM Multimedia*, Singapore, 2005.

[37] T. Westerveld, R. Cornacchia, J. van Gemert, D. Hiemstra, and A. de Vries. An integrated approach to text and image retrieval – the lowlands team at TRECVID 2005. In *Proc. of the TRECVID Workshop*, Gaithersburg, USA, 2005.

[38] X. Zhou and T. Huang. Relevance feedback in image retrieval: a comprehensive overview. *Multimedia Systems*, 8(6):536–544, 2003.