
Semantic Video Indexing*

Cees G.M. Snoek, Marcel Worring, Jan-Mark Geusebroek, Dennis C. Koelma, Frank J. Seinstra, and Arnold W.M. Smeulders

University of Amsterdam

8.1 Introduction

Query-by-keyword is the paradigm on which machine-based text search is still based. Elaborating on the success of text-based search engines, query-by-keyword also gains momentum in multimedia retrieval. For multimedia archives it is hard to achieve access, however, when based on text alone. Multimodal indexing is essential for effective access to video archives. For the automatic detection of specific concepts, the state-of-the-art has produced sophisticated and specialized indexing methods. Other than their textual counterparts, generic methods for semantic indexing in multimedia are neither generally available, nor scalable in their computational needs, nor robust in their performance. As a consequence, semantic access to multimedia archives is still limited. Therefore, there is a case to be made for a new approach to semantic video indexing.

The main problem for any semantic video indexing approach is the semantic gap between data representation and their interpretation by humans, as identified by Smeulders et al. [32]. In efforts to reduce the semantic gap, many video indexing approaches focus on specific semantic concepts with a small intra-class and large inter-class variability of content. Typical concepts and their detectors are *sunsets* by Smith and Chang [33] and the work by Zhang et al. on *news anchors* [43]. These concepts have become icons for video indexing. Although they have aided in achieving progress, this approach is limited when considering the plethora of concepts waiting to be detected. It is simply impossible to bridge the semantic gap by designing a tailor-made solution for each concept.

In this chapter we present a generic semantic video indexing method, which builds on the observation that produced video is the result of an authoring process. When producing a video, an author departs from a conceptual idea.

* © 2006 IEEE. Reprinted, with permission, from *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1678–1689, October 2006 [38].

The semantic intention is then articulated in (sub) consciously selected conventions and techniques for the purpose of emphasizing aspects of the content. The intention is communicated in context to the audience by a set of commonly shared notions. We aim to link the knowledge of years of media science research to semantic video analysis, see for example Boggs and Petrie [7] and Bordwell and Thompson [9]. We use the authoring-driven process of video production as the leading principle for generic video indexing.

Viewing semantic video indexing from an authoring perspective has the advantage that the most successful existing video indexing methods may be combined in one architecture. We first consider the vast amount of work performed in developing detection methods for specialized concepts [2, 5, 16, 18, 33, 43, 41]. If we measure the success of these methods in terms of benchmark detection performance, Informedia [18, 41] stands out. They focus on combining techniques from computer vision, speech recognition, natural language understanding, and artificial intelligence into a video indexing and retrieval environment. This has resulted in a large set of isolated and specialized concept detectors [18]. We build our generic indexing approach in part on the outputs of their detectors, but we do not use them in isolation.

In comparison to specialized detection methods, generic semantic indexing is rare. We discuss three successful examples of generic semantic indexing approaches [3, 13, 39]. Firstly, Fan et al. [13] propose the *ClassView* framework. The framework combines hierarchical semantic indexing with hierarchical retrieval. At the lowest level, the framework supports indexing of shots into concepts based on a large set of low-level visual features. At the second level a Bayes classifier maps concepts to semantic clusters. By assigning shots to a hierarchy of concepts, the framework supports queries based on semantic and visual similarity. As the authors indicate, the framework will provide more meaningful results if it would support multimodal content analysis. We aim for generic semantic indexing also, but we include multimodal analysis from the beginning. Secondly, Amir et al. [3] propose a system for semantic indexing using a detection pipeline. The pipeline starts with feature extraction, followed by consecutive aggregations on features, multiple modalities, and concepts. The pipeline optimizes the result by rule-based post filtering. We interpret the success of the system by the fact that all modules in the pipeline select the best of multiple hypotheses, and the exhaustive use of machine learning. Moreover, the authors were among the first to recognize that semantic indexing profits substantially from context. We adopt and extend their ideas related to hypothesis selection, machine learning, and the use of context for semantic indexing. All of the above generic methods ignore the important influence of the video production style in the analysis process. In addition to content and context, we identify layout and capture in [39] as important factors for semantic indexing of produced video. We propose in [39] a generic framework for produced video indexing combining four sets of style detectors in an iterative semantic classifier. Results indicate that the method obtains high accuracy for rich semantic concepts, rich meaning that concepts

share many similarities in their video production process. The framework is less suited for concepts that are not stylized. In the current paper, we generalize the idea of using style for semantic indexing.

We propose a generic approach for semantic indexing, we call the *semantic pathfinder*. It combines the most successful methods for semantic video indexing [3, 18, 39, 41] into an integrated architecture. The design principle is derived from the video production process, covering notions of content, style, and context. The architecture is built on several detectors, multimodal analysis, hypothesis selection, and machine learning. The semantic pathfinder combines analysis steps at increasing levels of abstraction, corresponding to well-known facts from the study of film and television production [7, 9]. Its virtue is its ability to learn the best path, from all explored analysis steps, on a per-concept basis. To demonstrate the effectiveness of the semantic pathfinder, the semantic indexing experiments are evaluated within a case study, using 85 hours of broadcast news video [30, 31].

8.1.1 Relation to Other Chapters

Chapter 2 discussed the important issue of multimedia management using metadata standards. An overview of basic machine learning techniques for recognizing patterns in multimedia content was presented in Chapter 3. Chapters 4, 5, and 7 presented an in-depth coverage of unimodal media analysis approaches on text, image, and speech respectively. In this chapter we present a unifying view to automatic extraction of metadata from multimedia sources, specifically focusing on multimodal video analysis in combination with machine learning.

8.1.2 Outline

The organization of this chapter is as follows. First, we introduce the broadcast news case study in Section 8.2. We highlight the data set used and elaborate on the lexicon of concepts that we index in a generic fashion. Our system architecture for semantic video indexing is presented in Section 8.3. We discuss its general machine learning architecture and its successive analysis steps. We present results in Section 8.4.

8.2 A Case Study on Broadcast News Video

8.2.1 Multimedia Archive

We focus on news video as a case study to study the problem of generic semantic indexing. The archive of choice is composed of 184 hours of ABC World News Tonight and CNN Headline News and is recorded in MPEG-1 format. The training data contains approximately 120 hours covering the period of January until June 1998. The 2004 test data contains the remaining 64 hours, covering the period of October until December 1998. Together with this video archive, CLIPS-IMAG [26] provided a camera shot segmentation.

We evaluate our semantic indexing approach on this data set to demonstrate the effectiveness of the semantic pathfinder for semantic access to multimedia archives.

8.2.2 Concept Lexicon

Before we elaborate on the video indexing architecture, we first define a lexicon A_S of 32 semantic concepts. The lexicon is indicative for future efforts to detect as much as 1000 concepts [17]. At present, it serves as a non-trivial illustration of concept possibilities. The semantic concept lexicon consists of the following concepts:

$A_S = \{ \textit{airplane take off}, \textit{American football}, \textit{animal}, \textit{baseball}, \textit{basket scored}, \textit{beach}, \textit{bicycle}, \textit{Bill Clinton}, \textit{boat}, \textit{building}, \textit{car}, \textit{cartoon}, \textit{financial news anchor}, \textit{golf}, \textit{graphics}, \textit{ice hockey}, \textit{Madeleine Albright}, \textit{news anchor}, \textit{news subject monologue}, \textit{outdoor}, \textit{overlayed text}, \textit{people}, \textit{people walking}, \textit{physical violence}, \textit{road}, \textit{soccer}, \textit{sporting event}, \textit{stock quotes}, \textit{studio setting}, \textit{train}, \textit{vegetation}, \textit{weather news} \}$.

Instantiations of the concepts in the lexicon are portrayed in Figure 8.1. The lexicon contains both general concepts, like *building*, *boat*, and *outdoor*, as well as specific concepts such as *news subject monologue* and *people walking*. We aim to detect all 32 concepts with the proposed system architecture.

8.3 Semantic Pathfinder

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



Fig. 8.1. Instances of the 32 concepts in the lexicon, which we aim to detect with the semantic pathfinder.

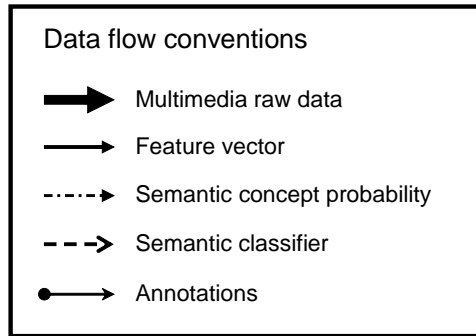


Fig. 8.2. Data flow conventions as used in this chapter. Different arrows indicate difference in data flows.

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 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.

The analysis steps in the semantic pathfinder exploit a common architecture, with a standardized input-output model, to allow for semantic integration. The conventions to describe the system architecture are indicated in Figure 8.2. An overview of the semantic pathfinder is given in Figure 8.3.

8.3.1 Analysis Step General Architecture

We perceive semantic indexing in video as a pattern recognition problem. We first need to segment a video. We opt for camera shots, indicated by i , following the standard in literature. Given pattern x , part of a shot, the aim is to detect a semantic concept ω from shot i using probability $p(\omega|x_i)$. Each analysis step in the semantic pathfinder extracts x_i from the data, and exploits a learning module to learn $p(\omega|x_i)$ for all ω in the semantic lexicon A_S . 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. This division prevents overtraining of the classifier. The general architecture for supervised learning in each analysis step is illustrated in Figure 8.4.

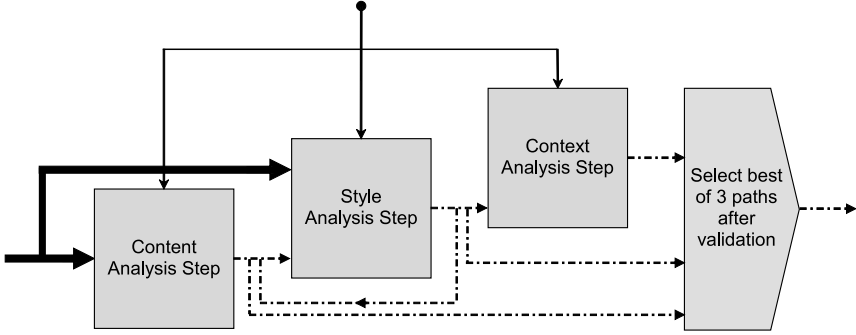


Fig. 8.3. The semantic pathfinder for one concept, using the conventions of Figure 8.2.

Supervised learning requires labeled examples. In part, we rely on the ground truth, which accompanies our news video data, provided by [20]. We remove the many errors from this manual annotation effort. It is extended to arrive at an incomplete, but reliable ground truth for all concepts in lexicon A_S . We split the training data a priori into a non-overlapping training set and validation set to prevent overfitting of classifiers in the semantic pathfinder. It should be noted that a reliable validation set would ideally require an as large as possible percentage of positively labeled examples, which is comparable to the training set. In practice this may be hard to achieve, however, as some

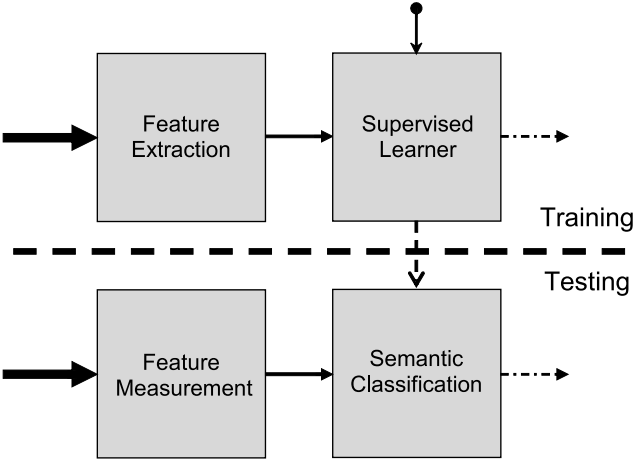


Fig. 8.4. General architecture of an analysis step in the semantic pathfinder, using the conventions of Figure 8.2.

concepts are sparse. The training set we use contains 85% of the training data, the validation set contains the remaining 15%. We summarize the percentage of positively annotated examples for each concept in training and validation set in Table 8.1.

We choose from a large variety of supervised machine learning approaches to obtain $p(\omega|x_i)$. 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 [12, 40] has proven to be a solid choice [3, 35]. In this framework each pattern x is represented in an n -dimensional space, spanned by extracted features. Within this feature space an optimal hyperplane is searched that separates it into two different categories, where the categories are represented by +1 and -1 respectively. The hyperplane has the following form: $\omega|(\mathbf{w} \cdot x + b)| \geq 1$, where \mathbf{w} is a weight vector, and b is a threshold. A hyperplane is considered optimal when the distance to the closest training examples is maximum for both categories. This distance is called the margin, see the example in Figure 8.5. The problem of finding the optimal hyperplane is a quadratic programming problem of the following form [40]:

$$\min_{\mathbf{w}, \xi} \left\{ \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \left(\sum_{i=1}^l \xi_i \right) \right\}, \quad (8.1)$$

under the following constraints:

$$\omega|(\mathbf{w} \cdot x_i + b)| \geq 1 - \xi_i, \quad \text{for } i = 1, 2, \dots, l, \quad (8.2)$$

where C is a parameter that allows us to balance training error and model complexity, l is the number of shots in the training set, and ξ_i are slack variables that are introduced when the data is not perfectly separable. These slack variables are useful when analyzing multimedia, since results of individual feature detectors typically include a number of false positives and negatives. The usual SVM method provides a margin, $\gamma(x_i)$, in the result. We prefer Platt's conversion method [25] to achieve a posterior probability of the result. It is defined as:

$$p(\omega|x_i) = \frac{1}{1 + \exp(\alpha\gamma(x_i) + \beta)}, \quad (8.3)$$

where the parameters α and β are maximum likelihood estimates based on training data. SVM classifiers thus trained for ω , result in an estimate $p(\omega|x_i, \mathbf{q})$, where \mathbf{q} are parameters of the SVM yet to be optimized.

The influence of the SVM parameters on concept detection is significant [22]. 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, \mathbf{q}^* . Here we use a three-fold cross-validation [19] to prevent overfitting of parameters. The result of

Table 8.1. Semantic concepts and the percentage of positively labeled examples used for the training set and the validation set.

<i>Semantic concept</i>	<i>Training (%)</i>	<i>Validation (%)</i>
Weather news	0.51	0.43
Stock quotes	0.26	0.30
News anchor	3.91	3.99
Overlaid text	0.26	0.17
Basket scored	1.07	0.97
Graphics	1.06	1.05
Baseball	0.74	0.66
Sporting event	2.27	2.44
People walking	1.92	1.97
Financial news anchor	0.35	0.35
Ice hockey	0.36	0.47
Cartoon	0.60	0.73
Studio setting	4.94	4.65
Physical violence	2.73	3.14
Vegetation	1.60	1.59
Boat	0.55	0.45
Golf	0.14	0.25
People	3.89	3.99
American football	0.05	0.10
Outdoor	7.52	8.60
Car	1.57	2.10
Bill Clinton	0.97	1.41
News subject monologue	3.84	3.96
Animal	1.35	1.34
Road	1.44	1.98
Beach	0.42	0.61
Train	0.21	0.36
Madeleine Albright	0.18	0.02
Building	4.95	4.81
Airplane take off	0.89	0.87
Bicycle	0.28	0.27
Soccer	0.06	0.09

the parameter search over \mathbf{q} is the improved model $p(\omega|x_i, \mathbf{q}^*)$, contracted to $p^*(\omega|x_i)$.

This concludes the introduction of the general architecture of all analysis steps in the semantic pathfinder.

8.3.2 Content Analysis Step

We view 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 textual features

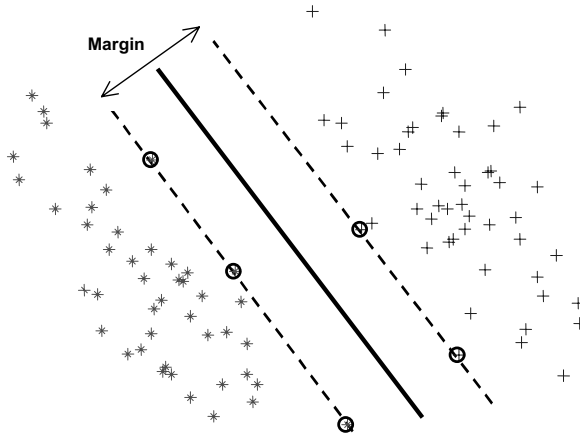


Fig. 8.5. Visual representation of the support vector machine framework. Here a two-dimensional feature space consisting of two categories is visualized. The solid bold line is chosen as optimal hyperplane because of the largest possible margin. The circled data points closest to the optimal hyperplane are called the support vectors.

obtained from transcribed speech and visual features. After modality specific data processing, we combine features in a multimodal representation. The data flow in the content analysis step is illustrated in Figure 8.6.

Visual Analysis

In the visual modality, we aim for segmentation of an image frame f into regional visual concepts. Ideally, a segmentation method should result in a precise partitioning of f according to the object boundaries, referred to as strong segmentation. However, weak segmentation, where f is partitioned into internally homogenous regions within the boundaries of the object, is often the best one can hope for [32]. We obtain a weak segmentation based on a set of visual feature detectors. Prior to segmentation we remove the border of each frame. The basis of feature extraction in the visual modality is weak segmentation.

Invariance was identified in [32] as a crucial aspect of a visual feature detector, e.g., to design features which limit the influence of accidental recording circumstances. We use color invariant visual features [15] to arrive at weak segmentation. The invariance covers the photometric variation due to shadow and shading, and geometrical variation due to scale and orientation. This invariance is needed as the conditions under which semantic concepts appear in large multimedia archives may vary greatly.

The feature extraction procedure we adhere to, computes per pixel a number of invariant features in vector \mathbf{u} . This vector then serves as the input for

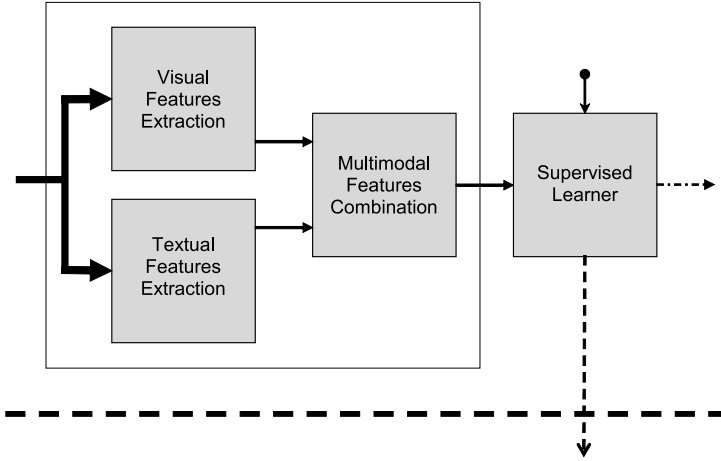


Fig. 8.6. Feature extraction and classification in the content analysis step, special case of Figure 8.4.

a multiclass SVM [12] that associates each pixel to one of the regional visual concepts defined in a visual concept lexicon Λ_V , using a labeled training set. Based on Λ_S , we define the following set of regional visual concepts:

$$\Lambda_V = \{ \text{colored clothing, concrete, fire, graphic blue, graphic purple, graphic yellow, grassland, greenery, indoor sport court, red carpet, sand, skin, sky, smoke, snow/ice, tuxedo, water body, wood} \}.$$

As we use invariant features, only a few examples per visual concept class are needed; in practice less than 10 per class. This pixel-wise classification results in the image vector \mathbf{w}_f , where \mathbf{w}_f contains one component per regional visual concept, indicating the percentage of pixels found for this class. Thus, \mathbf{w}_f is a weak segmentation of frame f in terms of regional visual concepts from Λ_V , see Figure 8.7 for an example segmentation.

We use Gaussian color measurements to obtain \mathbf{u} for weak segmentation [15]. We decorrelate RGB color values by linear transformation to the opponent color system [15]:

$$\begin{bmatrix} E \\ E_\lambda \\ E_{\lambda\lambda} \end{bmatrix} = \begin{pmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.35 \\ 0.34 & -0.6 & 0.17 \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{8.4}$$

Smoothing these values with a Gaussian filter, $G(\sigma)$, suppresses acquisition and compression noise. Moreover, we extract texture features by applying Gaussian derivative filters. We vary the size of the Gaussian filters, $\sigma = \{1, 2, 3.5\}$, to obtain a color representation that is compatible with variations in the target object size (leaving out pixel position parameters):

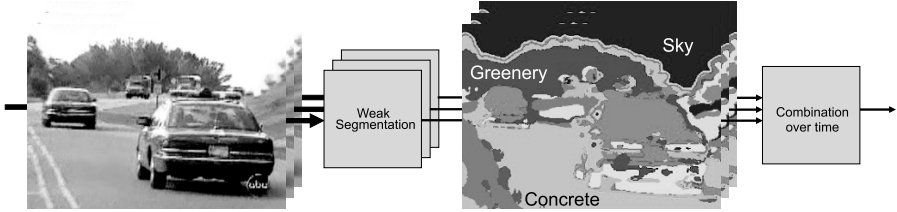


Fig. 8.7. Computation of the visual features, see Figure 8.6, is based on weak segmentation of an image frame into regional visual concepts. A combination over time is used to select one frame as representative for the shot.

$$\hat{E}_j(\sigma) = G_j(\sigma) * E, \quad \hat{E}_{\lambda j}(\sigma) = G_j(\sigma) * E_{\lambda}, \quad \hat{E}_{\lambda\lambda j}(\sigma) = G_j(\sigma) * E_{\lambda\lambda}, \quad (8.5)$$

where $j \in \{\emptyset, x, y\}$ indicates either spatial smoothing or spatial differentiation and that from now on the hat symbol ($\hat{\cdot}$) implies a dependence on σ . Normalizing each opponent color value by its intensity suppresses global intensity variations. This results in two chromaticity values per color pixel:

$$\hat{C}_{\lambda} = \frac{\hat{E}_{\lambda}}{\hat{E}}, \quad \hat{C}_{\lambda\lambda} = \frac{\hat{E}_{\lambda\lambda}}{\hat{E}}. \quad (8.6)$$

Furthermore, we obtain rotationally invariant features by taking Gaussian derivative filters and combining the responses into two chromatic gradients:

$$\hat{C}_{\lambda w} = \sqrt{\hat{C}_{\lambda x}^2 + \hat{C}_{\lambda y}^2}, \quad \hat{C}_{\lambda\lambda w} = \sqrt{\hat{C}_{\lambda\lambda x}^2 + \hat{C}_{\lambda\lambda y}^2}, \quad (8.7)$$

where $\hat{C}_{\lambda x}$, $\hat{C}_{\lambda y}$, $\hat{C}_{\lambda\lambda x}$, and $\hat{C}_{\lambda\lambda y}$ are defined as:

$$\begin{aligned} \hat{C}_{\lambda x} &= \frac{\hat{E}_{\lambda x} \hat{E} - \hat{E}_{\lambda} \hat{E}_x}{\hat{E}^2}, & \hat{C}_{\lambda\lambda x} &= \frac{\hat{E}_{\lambda\lambda x} \hat{E} - \hat{E}_{\lambda\lambda} \hat{E}_x}{\hat{E}^2}, \\ \hat{C}_{\lambda y} &= \frac{\hat{E}_{\lambda y} \hat{E} - \hat{E}_{\lambda} \hat{E}_y}{\hat{E}^2}, & \hat{C}_{\lambda\lambda y} &= \frac{\hat{E}_{\lambda\lambda y} \hat{E} - \hat{E}_{\lambda\lambda} \hat{E}_y}{\hat{E}^2}. \end{aligned} \quad (8.8)$$

The seven measurements computed in (8.5)–(8.7), and each calculated over three scales, yield a 21-dimensional invariant feature vector \mathbf{u} per pixel.

Segmenting image frames into regional visual concepts at the granularity of a pixel is computationally intensive. We estimate that the processing of the entire case study data set would have taken around 250 days on the fastest sequential machine available to us. As a first reduction of the analysis load, we analyze 1 out of 15 frames only. For the remaining image processing effort we apply the Parallel-Horus software architecture [29]. This architecture, consisting of a large collection of low-level image processing primitives, allows the programmer to write sequential applications with efficient parallel execution on commonly available commodity clusters. Application of Parallel-Horus, in combination with a distributed cluster consisting of 200 dual 1-GHz Pentium-III CPUs [6], reduced the processing time to less than 60 hours [29].

The features over time are combined into one vector for the shot i . Averaging over individual frames is not a good choice, as the visual representation should remain intact. Instead, we opt for a selection of the most representative frame or visual vector. To decide which f is the most representative for i , weak segmented image \mathbf{w}_f is the input for an SVM that computes a probability $p^*(\omega|\mathbf{w}_f)$. We select \mathbf{w}_f that maximizes the probability for a concept from A_S within i , given as:

$$\mathbf{v}_i = \arg \max_{f \in f_i} p^*(\omega|\mathbf{w}_f). \quad (8.9)$$

The visual vector \mathbf{v}_i , containing the best weak segmentation, is the final result of the visual analysis.

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. From the text we remove the frequently occurring stopwords. After stop-word 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, A_T^ω , as this uttered word lexicon is specific for that concept. We modify the procedure for Person X concepts, i.e., *Madeleine Albright* and *Bill Clinton*, to optimize results. In broadcast news, a news anchor or reporter mentions names or other indicative words just before or after a person is visible. To account for this observation, we stretch the shot boundaries with five seconds on each side for Person X concepts. For these concepts, this procedure assures that the textual feature analysis considers even more textual content. For feature extraction we compare the text associated with each shot with A_T^ω . This comparison yields a text vector \mathbf{t}_i for shot i , which contains the histogram of the words in association with ω .

Multimodal Analysis and Classification

The result of the content analysis step is a multimodal vector \mathbf{m}_i that integrates all unimodal results. We concatenate the visual vector \mathbf{v}_i with the text vector \mathbf{t}_i , to obtain \mathbf{m}_i . After this modality fusion, \mathbf{m}_i serves as the input for the supervised learning module. To optimize parameter settings, we use three-fold cross-validation on the training set. The content analysis step associates probability $p^*(\omega|\mathbf{m}_i)$ with a shot i , for all ω in A_S .

8.3.3 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 [39, 34],

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 Figure 8.8. Note that in contrast to the content analysis step, where we learn specific content features from a data set, content features in the style analysis step are generic and independent of the data set.

Style Analysis

We develop detectors for all four production roles as feature extraction in the style analysis step. Each style detector uses an existing software implementation as a basis. The output of such a base detector is then aggregated and synchronized to a camera shot. We categorize the resulting production-derived features based on experimentally obtained thresholds. Together, these three components define a style detector. We refer to our previous work for specific implementation details of the detectors [34, Appendix A],[39]. We have chosen to convert the output of all style detectors to an ordinal scale, as this allows for easy 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 [36]. Overlaid text is another informative descriptor. Its presence is detected by a text localization algorithm [27]. To segment the auditory layout, periods of speech and silence are detected based on an automatic speech recognition system [14]. We obtain a voice-over detector by combining the speech segmentation with the camera shot segmentation [39]. The set of layout features is thus given by: $\mathcal{L} = \{\textit{shot length}, \textit{overlaid text}, \textit{silence}, \textit{voice-over}\}$.

As concerns the content \mathcal{C} , a frontal face detector [28] is applied to detect people. We count the number of faces, and for each face its location is derived [39]. Apart from faces, we also detect the presence of cars [28]. In addition, we measure the average amount of object motion in a camera shot [35]. Based on speaker identification [14] we identify each of the three most frequent speakers. The camera shot is checked for the presence on the basis of speech from one of the three [39]. The length of text strings recognized by Video Optical Character Recognition [27] is used as a feature [39]. In addition, the strings are used as input for a named entity recognizer [41]. On the transcribed text obtained by the LIMSI automatic speech recognition system [14], we also apply named entity recognition. The set of content features is thus given by: $\mathcal{C} = \{\textit{faces}, \textit{face location}, \textit{cars}, \textit{object motion}, \textit{frequent speaker}, \textit{overlaid text length}, \textit{video text named entity}, \textit{voice named entity}\}$.

For capture \mathcal{T} , we compute the camera distance from the size of detected faces [28, 39]. It is undefined when no face is detected. In addition to camera distance, several types of camera work are detected [4], e.g., pan, tilt, zoom, and so on. Finally, for capture we also estimate the amount of camera motion [4]. The set of capture features is thus given by: $\mathcal{T} = \{\textit{camera distance}, \textit{camera work}, \textit{camera motion}\}$.

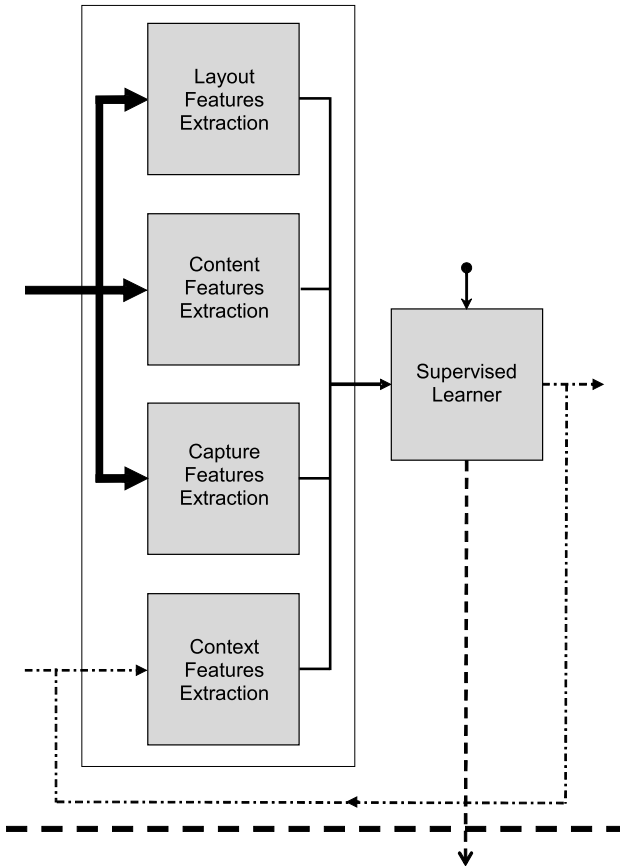


Fig. 8.8. Feature extraction and classification in the style analysis step, special case of Figure 8.4.

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. As initial concept we detect news reporters. We recognize news reporters by edit distance matching of strings, obtained from the transcript and video text, with a database of names of CNN and ABC affiliates [39]. The other concepts that are added to the context stem from \mathcal{A}_S . To prevent bias from domain knowledge, we use

the performance on the validation set of all concepts from A_S in the content analysis step as the ordering for the context. For this ordering we again refer to Table 8.1. To assign detection results for the first and least difficult concept, $\omega_1 = \textit{weather news}$, we rank all shot results on $p_i^*(\omega_1|\mathbf{m}_i)$. 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: $\mathcal{S} = \{\textit{news reporter}, \textit{content analysis step } \omega_1\}$.

The concatenation of $\{\mathcal{L}, \mathcal{C}, \mathcal{T}, \mathcal{S}\}$ for shot i yields the style vector \mathbf{s}_i . This vector forms the input for an iterative classifier that trains a style model for each concept in lexicon A_S .

Iterative Style Classification

We start from an ordering of concepts in the context, as defined above. The iteration of the classifier begins with concept ω_1 . After concatenation with the other style features this yields $\mathbf{s}_{i,1}$ the first style vector of the first iteration. $\mathbf{s}_{i,1}$ contains the combined results of the content analysis step and the style analysis step. We classify ω_1 again based on $\mathbf{s}_{i,1}$. This yields the posterior probability $p^*(\omega_1|\mathbf{s}_{i,1})$. When $p^*(\omega|\mathbf{s}_i) \geq \delta$ the concept ω_1 is considered present in the style representation, else it is considered absent. The threshold δ is set a priori at a fixed value of 0.5. In this process the classifier replaces the feature for concept ω_1 , from the content analysis step, by the new feature ω_1^+ . The style analysis step adds more aspects of the author influence to the results obtained with the content analysis step. In the next iteration of the classification procedure, the classifier adds $\omega_2 = \textit{stock quotes}$ from the content analysis step to the context. This yields $\mathbf{s}_{i,2}$. As explained above, the classifier replaces the ω_2 feature from the content analysis step by the styled version ω_2^+ based on $p^*(\omega_2|\mathbf{s}_{i,2})$. This iterative process is repeated for all ω in lexicon A_S .

We classify all ω in A_S again in the style analysis step. As the result of the content analysis step is only one of the many features in our style vector representation in the style analysis step, we also use three-fold cross-validation on the training set to optimize parameter settings in this analysis step. We use the resulting probability as output for concept detection in the style analysis step. In addition, it forms the input for the next analysis step in our semantic pathfinder.

8.3.4 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.

Semantic Analysis

The style analysis step yields a probability for each shot i and all concepts ω in A_S . The probability indicates whether a concept is present. We use the 32

concept scores as semantic features. We fuse them into context vector \mathbf{c}_i , see Figure 8.9.

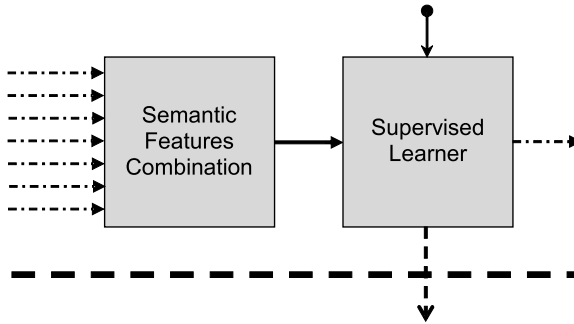


Fig. 8.9. Feature extraction and classification in the context analysis step, special case of Figure 8.4.

From \mathbf{c}_i we learn relations between concepts automatically. To that end, \mathbf{c}_i serves as the input for a supervised learning module, which associates a contextual probability $p^*(\omega|\mathbf{c}_i)$ to a shot i for all ω in A_S . To optimize parameter settings, we use three-fold cross-validation on the previously unused data from the validation set.

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 a probability based on content, style, and context. We select from the three possibilities the one that maximizes average precision based on validation set performance. The semantic pathfinder provides us with the opportunity to decide whether a one-shot analysis step is best for the concept only concentrating on 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 path using content, style, and context.

8.4 Indexing Results on 32 Semantic Concepts

We evaluated detection results for all 32 concepts in each analysis step. Given the already enormous size of the data sets and the large amounts of annotation – yet limited in terms of completeness – we have performed one pass for 32 concepts through the entire semantic pathfinder. We report the *precision at 100*, which indicates the number of correct shots within the first 100 results in Table 8.2.

Table 8.2. Test set precision at 100 after the three steps, for a lexicon of 32 concepts. The best result is given in bold. The corresponding path is selected in the semantic pathfinder.

<i>Semantic concept</i>	<i>Content analysis step</i>	<i>Style analysis step</i>	<i>Context analysis step</i>	<i>Semantic pathfinder</i>
News subject monologue	0.55	1.00	1.00	1.00
Weather news	1.00	1.00	1.00	1.00
News anchor	0.98	0.98	0.99	0.99
Overlaid text	0.84	0.99	0.93	0.99
Sporting event	0.77	0.98	0.93	0.98
Studio setting	0.95	0.96	0.98	0.98
Graphics	0.92	0.90	0.91	0.91
People	0.73	0.78	0.91	0.91
Outdoor	0.62	0.83	0.90	0.90
Stock quotes	0.89	0.77	0.77	0.89
People walking	0.65	0.72	0.83	0.83
Car	0.63	0.81	0.75	0.75
Cartoon	0.71	0.69	0.75	0.75
Vegetation	0.72	0.64	0.70	0.72
Ice hockey	0.71	0.68	0.60	0.71
Financial news anchor	0.40	0.70	0.71	0.70
Baseball	0.54	0.43	0.47	0.54
Building	0.53	0.46	0.43	0.53
Road	0.43	0.53	0.51	0.51
American football	0.46	0.18	0.17	0.46
Boat	0.42	0.38	0.37	0.37
Physical violence	0.17	0.25	0.31	0.31
Basket scored	0.24	0.21	0.30	0.30
Animal	0.37	0.26	0.26	0.26
Bill Clinton	0.26	0.35	0.37	0.26
Golf	0.24	0.19	0.06	0.24
Beach	0.13	0.12	0.12	0.12
Madeleine Albright	0.12	0.05	0.04	0.12
Airplane take off	0.10	0.08	0.08	0.08
Bicycle	0.09	0.08	0.07	0.08
Train	0.07	0.07	0.03	0.07
Soccer	0.01	0.01	0.00	0.01
<i>Mean</i>	<i>0.51</i>	<i>0.53</i>	<i>0.54</i>	<i>0.57</i>

We observe from the results that the learned best path (printed in bold) indeed varies over the concepts. The virtue of the semantic pathfinder is demonstrated by the fact that for 12 concepts, the learning phase indicates it is best to concentrate on content only. For five concepts, the semantic pathfinder demonstrates that a two-step path is best (where in 15 cases addition of style features has a marginal positive or negative effect). For 15 concepts, the context analysis step obtains a better result. Context aids substantially in the

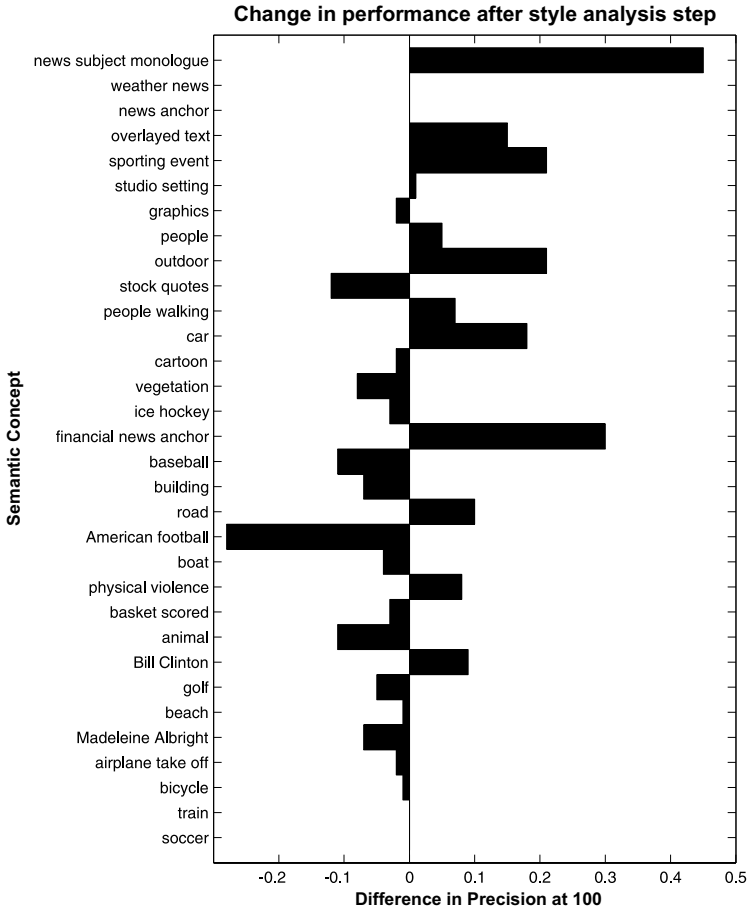


Fig. 8.10. Influence of the style analysis step on precision at 100 performance for a lexicon of 32 semantic concepts. Note a considerable decrease (American football) or increase (news subject monologue) in performance when adding production style information.

performance for five concepts. As an aside we note that the precision at 100, when averaged over all concepts, steadily increases from 0.51 to 0.57 while traversing the different semantic analysis paths.

The results demonstrate the virtue of the semantic pathfinder. Concepts are divided by the analysis step after which they achieve best performance. Some concepts are just content, style does not affect them. In such cases as *American football* there is style-wise too much confusion with other sports to add new value in the path. Shots containing *stock quotes* suffer from a similar problem. Here false positives contain many stylistically similar results like graphical representations of survey and election results. For complex con-

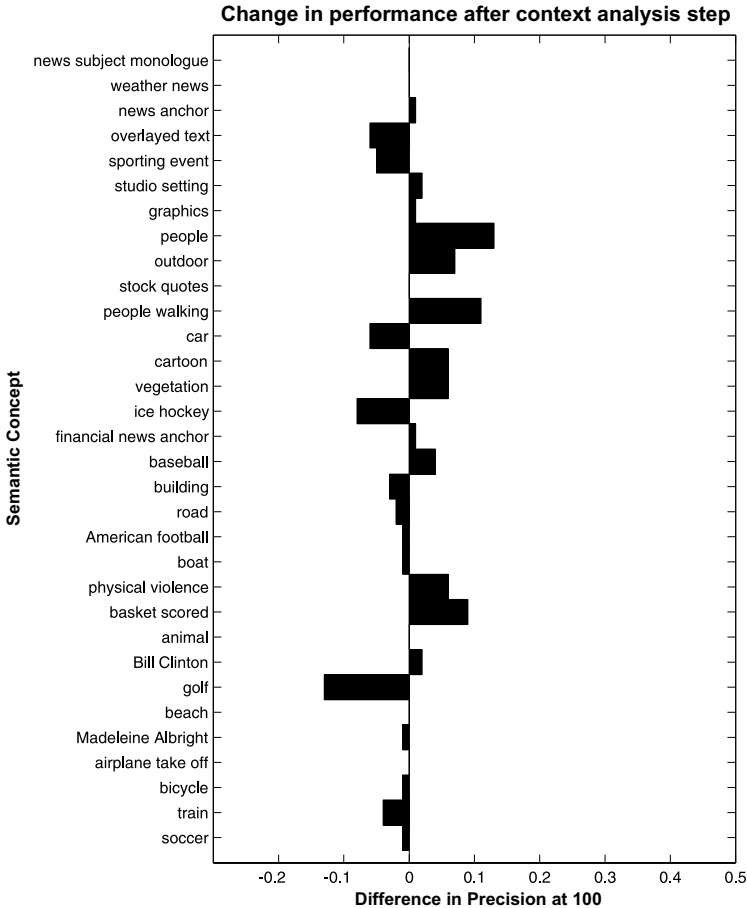


Fig. 8.11. Influence of the context analysis step on precision at 100 performance for a lexicon of 32 semantic concepts. Note a considerable decrease (golf) or increase (people) in performance when adding context information.

cepts, analysis based on content and style is not enough. They require the use of context. The context analysis step is especially good in detecting named events, like *people walking*, *physical violence*, and *basket scored*. The results offer us the possibility to categorize concepts according to the analysis step of the semantic pathfinder that yields the best performance.

The content analysis step seems to work particularly well for semantic concepts that have a small intra-class variability of content: *weather news* and *news anchor* for example. In addition, this analysis step aids in detection of accidental content like *building*, *vegetation*, *bicycle*, and *train*. However, for some of those concepts, e.g., *bicycle* and *train*, the performance is still disappointing. Another observation is that when one aims to distinguish subgenres,

e.g., *ice hockey*, *baseball*, and *American football*, the content analysis step is the best choice.

After the style analysis step, we obtain an increase in performance for 12 concepts, see Figure 8.10. Especially when the concepts are semantically rich: e.g., *news subject monologue*, *financial news anchor*, and *sporting event*, the style helps. As expected, index results in the style analysis step improve on the content analysis step when style is a distinguishing property of the concept and degrade the result when similarity in style exists between different concepts.

Results after the context analysis step in Figure 8.11 show that performance increases for 13 concepts. The largest positive performance difference between the context analysis step and the style analysis step occurs for concept *people*. Concept *people* profits from sport-related concepts like *baseball*, *basket scored*, *American football*, *ice hockey*, and *sporting event*. In contrast, *golf* suffers from detection of *outdoor* and *vegetation*. When we detect *golf*, these concepts are also present frequently. The inverse, however, is not necessarily the case, i.e., when we detect *outdoor* it is not necessarily on a golf course. Based on these observations we conclude that, apart from named events, detection results of the context analysis step are similar to those of the style analysis step. Index results improve based on presence of semantically related concepts, but the context analysis step is unable to capture the semantic structure between concepts and for some concepts, this is leading to a drop in performance.

The above results show that the semantic pathfinder facilitates generic video indexing. In addition, the semantic pathfinder provides the foundation of a technique taxonomy for solving semantic concept detection tasks. The fact that subgenres like *ice hockey*, *golf*, and *American football* behave similarly indicate the predictive value of the pathfinder for other subgenres. The same holds for semantically rich concepts like *news subject monologue*, *financial news anchor*, and *sporting event*. We showed that for named events, such as *basket scored*, *physical violence*, and *people walking*, one should apply a detector that is based on the entire semantic pathfinder. The significance of the semantic pathfinder is its generalizing power combined with the fact that addition of new information in the analysis can be considered by concept type.

8.4.1 Usage Scenarios

The results from the semantic pathfinder facilitate the development of various applications. The lexicon of 32 semantic concepts allows for querying a video archive by concept. Elsewhere [37] we combined into the *MediaMill* semantic video search engine query-by-concept, query-by-keyword, query-by-example, and interactive filtering, see Figure 8.12. In addition to interactive search, the set of indexes is also applicable in a personalized retrieval setting. A feasible scenario is that users with a specific interest in sports are provided with personalized summaries when and where they need it. The sketched applications provide a semantic access to multimedia archives.

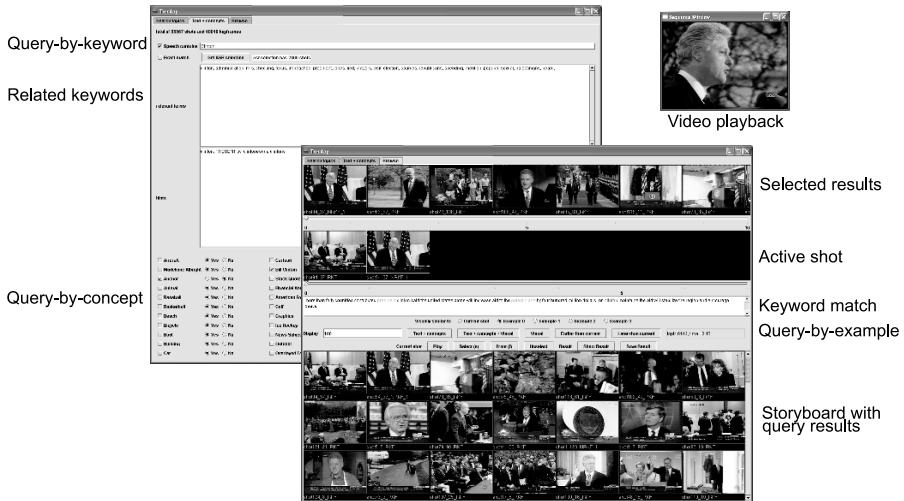


Fig. 8.12. Interface of the MediaMill semantic video search engine. The system allows for interactive query-by-concept using 32 concepts. In addition, it facilitates query-by-similarity in the form of query-by-keyword, and query-by-example. Results are presented in a storyboard.

8.5 Summary

In this chapter, we present the semantic pathfinder for semantic access to multimedia archives. The semantic pathfinder is a generic approach for video indexing. It is based on the observation that produced video is the result of an authoring process. The semantic pathfinder exploits the authoring metaphor in an effort to bridge the semantic gap. The architecture is built on a variety of detector types, multimodal analysis, hypothesis selection, and machine learning. The semantic pathfinder selects the best path through content analysis, style analysis, and context analysis. After machine learning it appears that the analysis is completed after content analysis only when concepts share many similarities in their multimodal content. It appears also that the semantic path runs up to style analysis when the professional habits of television are evident to the concept. Finally, it exploits a path based on content, style, and context for concepts that are primarily intentional, see Table 8.2 and Figures 8.10 and 8.11.

Experiments with a lexicon of 32 semantic concepts demonstrate that the semantic pathfinder allows for generic video indexing, while confirming the value of the authoring metaphor in indexing. In addition, the results over the various analysis steps indicate that a technique taxonomy exists for solving semantic concept detection tasks; depending on whether content, style, or context is most suited for indexing. For some concepts the precision at 100 performance is still quite low. For selecting illustrative footage, this may already be sufficient. This is not yet so for tasks that require accurate retrieval.

However, the trend in results over the past years indicates that automated search in video archives lures at the horizon.

8.6 Further Reading

Basic techniques for video indexing are discussed in the review papers by Bolle et al. [8] and Brunelli et al. [10]. Smeulders et al. [32] present an in depth overview of content-based image retrieval. Where these papers emphasize the visual analysis in video indexing, the review paper of Wang et al. [42] stresses audio analysis. For an overview of text analysis methods we refer to the book by Manning and Schütze [21]. A broad introduction to multimodal semantic video indexing literature can be found in our previous work [36] and the work of Naphade and Huang [23].

Statistical pattern recognition is an indispensable tool for anyone working in semantic video indexing. An excellent introduction and overview is in the paper by Jain et al. [19]. At present, the support vector machine framework is the classifier of choice in the most successful semantic video indexing systems [1, 3, 38]. An in depth theoretical discussion on the support vector machine is in the book by its inventor Vapnik [40]. A more accessible tutorial is the paper by Burges [11].

For recent updates on the state-of-the-art in the field we refer to the proceedings of the yearly *ACM Multimedia Conference*, the *International Conference on Image and Video Retrieval*, and the *IEEE International Conference on Multimedia & Expo*. The most important journals in the field are *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Multimedia*, *IEEE Multimedia*, and *ACM Transactions on Multimedia Computing, Communications and Applications*.

We have deliberately left out the NIST TRECVID video retrieval benchmark in our discussion on semantic video indexing, as this benchmark is the topic of Chapter 13. The benchmark aims to promote progress in video retrieval via open, metrics-based evaluation [30, 31]. Tasks include camera shot segmentation, story segmentation, semantic concept detection, and several search tasks. Because of its widespread acceptance in the field, resulting in large participation of teams from both academic and corporate research labs worldwide, the benchmark can be regarded as the *de facto* standard to evaluate performance of semantic video indexing and retrieval research. The most recent developments in semantic video indexing are accessible via the electronic proceedings of the TREC workshop on Video Retrieval Evaluation [24].

References

1. W. H. Adams, G. Iyengar, C.-Y. Lin, M.R. Naphade, C. Neti, H.J. Nock, and J.R. Smith. Semantic indexing of multimedia content using visual, audio, and text cues. *EURASIP Journal on Applied Signal Processing*, (2):170–185, 2003.

2. A.A. Alatan, A.N. Akansu, and W. Wolf. Multimodal dialogue scene detection using hidden Markov models for content-based multimedia indexing. *Multimedia Tools Applicat.*, 14(2):137–151, 2001.
3. A. Amir, M. Berg, S.-F. Chang, W. Hsu, G. Iyengar, C.-Y. Lin, M.R. Naphade, A.P. Natsev, C. Neti, H.J. Nock, J.R. Smith, B.L. Tseng, Y. Wu, and D. Zhang. IBM research TRECVID-2003 video retrieval system. In *Proc. TRECVID Workshop*, NIST Special Publication, Gaithersburg, USA, 2003.
4. J. Baan, A. van Ballegooij, J.-M. Geusebroek, D. Hiemstra, J. den Hartog, J. List, C. Snoek, I. Patras, S. Raaijmakers, L. Todoran, J. Vendrig, A. de Vries, T. Westerveld, and M. Worring. Lazy users and automatic video retrieval tools in (the) lowlands. In E.M. Voorhees and D.K. Harman, editors, *Proc. 10th Text REtrieval Conference*, volume 500-250 of *NIST Special Publication*, Gaithersburg, USA, 2001.
5. N. Babaguchi, Y. Kawai, and T. Kitahashi. Event based indexing of broadcasted sports video by intermodal collaboration. *IEEE Trans. Multimedia*, 4(1):68–75, 2002.
6. H.E. Bal et al. The distributed ASCI supercomputer project. *Operating Syst. Review*, 34(4):76–96, 2000.
7. J.M. Boggs and D.W. Petrie. *The Art of Watching Films*. Mayfield Publishing Company, Mountain View, USA, 5th edition, 2000.
8. R.M. Bolle, B.-L. Yeo, and M.M. Yeung. Video query: Research directions. *IBM Journal of Research and Development*, 42(2):233–252, 1998.
9. D. Bordwell and K. Thompson. *Film Art: An Introduction*. McGraw-Hill, New York, USA, 5th edition, 1997.
10. R. Brunelli, O. Mich, and C.M. Modena. A survey on the automatic indexing of video data. *J. Visual Commun. Image Representation*, 10(2):78–112, 1999.
11. C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
12. C.-C. Chang and C.-J. Lin. *LIBSVM: a library for Support Vector Machines*, 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
13. J. Fan, A.K. Elmagarmid, X. Zhu, W.G. Aref, and L. Wu. *ClassView*: hierarchical video shot classification, indexing, and accessing. *IEEE Trans. Multimedia*, 6(1):70–86, 2004.
14. J.L. Gauvain, L. Lamel, and G. Adda. The LIMSI broadcast news transcription system. *Speech Commun.*, 37(1–2):89–108, 2002.
15. J.M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, and H. Geerts. Color invariance. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(12):1338–1350, 2001.
16. N. Haering, R. Qian, and I. Sezan. A semantic event-detection approach and its application to detecting hunts in wildlife video. *IEEE Trans. Circuits Syst. Video Technol.*, 10(6):857–868, 2000.
17. A.G. Hauptmann. Towards a large scale concept ontology for broadcast video. In *CIVR*, volume 3115 of *LNCS*, pages 674–675. Springer-Verlag, 2004.
18. A.G. Hauptmann, R.V. Baron, M.-Y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W.-H. Lin, T. Ng, N. Moraveji, N. Papernick, C.G.M. Snoek, G. Tzanetakis, J. Yang, R. Yang, and H.D. Wactlar. Informedia at TRECVID 2003: Analyzing and searching broadcast news video. In *Proc. TRECVID Workshop*, NIST Special Publication, Gaithersburg, USA, 2003.
19. A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(1):4–37, 2000.

20. C.-Y. Lin, B.L. Tseng, and J.R. Smith. Video collaborative annotation forum: Establishing ground-truth labels on large multimedia datasets. In *Proc. TRECVID Workshop*, NIST Special Publication, Gaithersburg, USA, 2003.
21. C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, USA, 1999.
22. M.R. Naphade. On supervision and statistical learning for semantic multimedia analysis. *J. Visual Commun. Image Representation*, 15(3):348–369, 2004.
23. M.R. Naphade and T.S. Huang. Extracting semantics from audiovisual content: The final frontier in multimedia retrieval. *IEEE Trans. Neural Networks*, 13(4):793–810, 2002.
24. NIST. TREC Video Retrieval Evaluation. <http://www-nlpir.nist.gov/projects/trecvid/>.
25. J.C. Platt. Probabilities for SV machines. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.
26. G.M. Quénot, D. Moraru, L. Besacier, and P. Mulhem. CLIPS at TREC-11: Experiments in video retrieval. In E.M. Voorhees and L.P. Buckland, editors, *Proc. 11th Text REtrieval Conference*, volume 500-251 of *NIST Special Publication*, Gaithersburg, USA, 2002.
27. T. Sato, T. Kanade, E.K. Hughes, M.A. Smith, and S. Satoh. Video OCR: Indexing digital news libraries by recognition of superimposed caption. *Multimedia Syst.*, 7(5):385–395, 1999.
28. H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *Int'l J. Comput. Vision*, 56(3):151–177, 2004.
29. F.J. Seinstra, C.G.M. Snoek, D. Koelma, J.M. Geusebroek, and M. Worring. User transparent parallel processing of the 2004 NIST TRECVID data set. In Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), pages 90–98, Denver, USA, 2005.
30. A.F. Smeaton, W. Kraaij, and P. Over. The TREC VIDEO retrieval evaluation (TRECVID): A case study and status report. In *Proc. RIAO 2004*, Avignon, France, 2004.
31. A.F. Smeaton, P. Over, and W. Kraaij. TRECVID: Evaluating the effectiveness of information retrieval tasks on digital video. In *Proceedings of the ACM MM'04 (Multimedia)*, pages 652–655, New York, USA, 2004.
32. A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(12):1349–1380, 2000.
33. J.R. Smith and S.-F. Chang. Visually searching the Web for content. *IEEE Multimedia*, 4(3):12–20, 1997.
34. C.G.M. Snoek. *The Authoring Metaphor to Machine Understanding of Multimedia*. PhD thesis, University of Amsterdam, 2005.
35. C.G.M. Snoek and M. Worring. Multimedia event-based video indexing using time intervals. *IEEE Trans. Multimedia*, 7(4):638–647, 2005.
36. C.G.M. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools Applicat.*, 25(1):5–35, 2005.
37. C.G.M. Snoek, M. Worring, J. van Gemert, J.M. Geusebroek, D. Koelma, G.P. Nguyen, O. de Rooij, and F. Seinstra. MediaMill: Exploring news video archives based on learned semantics. In *Proceedings of the ACM International Conference on Multimedia*, pages 225–226, Singapore, November 2005.

38. 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. Machine Intell.*, 28(10):1678–1689, 2006.
39. C.G.M. Snoek, M. Worring, and A.G. Hauptmann. Learning rich semantics from news video archives by style analysis. *ACM Trans. Multimedia Computing, Comm. Applications*, 2(2):91–108, 2006.
40. V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, USA, 2nd edition, 2000.
41. H.D. Wactlar, M.G. Christel, Y. Gong, and A.G. Hauptmann. Lessons learned from building a terabyte digital video library. *IEEE Computer*, 32(2):66–73, 1999.
42. Y. Wang, Z. Liu, and J. Huang. Multimedia content analysis using both audio and visual clues. *IEEE Signal Processing Magazine*, 17(6):12–36, 2000.
43. H.-J. Zhang, S.Y. Tan, S.W. Smoliar, and Y. Gong. Automatic parsing and indexing of news video. *Multimedia Syst.*, 2(6):256–266, 1995.