Adrian Ulges

Visual Concept Learning from User-tagged Web Video

Dissertation

genehmigt vom Fachbereich Informatik der Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)

Dekan:
Prof. Dr. Karsten Berns, Universität Kaiserslautern

Berichterstatter:
Prof. Dr. Thomas Breuel, Universität Kaiserslautern
Außerordentlicher Prof. Dr. Marcel Worring, Universität Amsterdam

Vorsitzender der Promotionskommission:
Prof. Dr. Markus Nebel, Universität Kaiserslautern

Datum der Aussprache:
9. Oktober 2009

D 386

Acknowledgements

This thesis would be incomplete without lots of Thank Yous to people who shaped it in multiple ways. First, I want to thank Prof. Thomas Breuel, who gave me the opportunity to work on my favorite PhD topic and was a neverending source of inspiring ideas. I also want to thank Prof. Marcel Worring, who provided new viewpoints and helped improving this work significantly. Christoph Lampert and Daniel Keysers deserve to be mentioned for their outstanding mentorship.

Next, I want to thank all the people at IUPR for numerous discussions, hints and fixes, for the first Wednesdays in a month, the PES battles, and — in general — for new perspectives on pattern recognition, life, the universe and everything. Thanks to all of you!

Particularly, I want to thank other contributors to this thesis: its volunteer reviewers — Armin Stahl, Damian Borth, Daniel Keysers, Faisal Shafait, Joost van Beusekom, and Oliver Wirjadi — and colleagues and students that contributed with their work — Manni Duan, Tsvetana Spasova, Christian Schulze, Damian Borth (again), and Markus Koch. Other researchers have provided data, software, and advice — Alexander Hauptmann, Chih-Chung Chang and Chih-Jen Lin, Herbert Bay, Krystian Mikolajczyk, Peter Gehler, Roberto Paredes, R. Manmatha and Shaolei Feng, and Vladimir Kolmogorov and Yuri Boykov.

Finally, I want to thank Rebecca — for loving me and for (still) being so patient with me — and my parents: Mama und Papa, ich möchte euch für euren Glauben und eure Unterstützung danken, und für zarte Erinnerungen an wichtigere Dinge.

Abstract

As digital video has become a source of information and entertainment to millions of users, video databases grow at enormous rates, and a need for new efficient indexing and search strategies has been recognized by research and industry. In this context, *concept detection* aims at a machine indexing by automatically linking video scenes with semantic concepts appearing in them.

Existing concept detection systems rely on manual annotation for concept learning, and are thus limited by the effort associated with training data acquisition. To overcome this problem, this thesis describes a concept learning approach that requires significantly less manual supervision compared to standard methods. To achieve this, user-tagged web video is employed (as offered by portals like YouTube). Four contributions are made that greatly enhance our ability to use this data source for training, regarding its content, label noise, context, and motion information.

To make use of web video content, this thesis presents a concept detection system that employs clips downloaded from YouTube as training data, with class labels being automatically derived from user-generated tags and descriptions. It is demonstrated on standard datasets from the TRECVID benchmark that the resulting detectors generalize comparably well to novel domains as detectors trained on manually acquired ground truth. At the same time, the approach offers a much more scalable and flexible way of concept learning.

To address label noise (i.e., the problem that user-generated tags are coarse, subjective, and context-dependent), this thesis proposes to adapt the statistical models underlying concept detection. Web tags are viewed as unreliable indicators of true label information, which is modeled as a latent random variable and inferred during concept detector training. This novel approach (called *relevance filtering*) is validated to improve concept learning from web video significantly compared to supervised standard methods, for both a generative and a discriminative base model.

To make use of context, user-generated category labels are employed, another valuable feature of web video. It is demonstrated that this information can be used by combining concept detection with *style modeling*: a distinct model is learned per category (or *style*, respectively) and used for an accurate concept detection. Test images are mapped to a style using their context (for example, other pictures taken at the same event). This approach is demonstrated to improve performance

by up to 100% on Flickr photos ($n = 32,000$). On the well-known COREL-5K image annotation benchmark, the proposed method gives a mean recall/precision of 39%/25%, which is the best result reported to date.

Finally, to make use of motion information, this thesis suggests to improve the learning and recognition of objects using motion-based segmentation. Two novel motion segmentation approaches are presented, one based on a globally optimal branch-and-bound search of parameter space, one on a combination of motion and color information. These approaches are integrated with a patch-based recognition method, achieving an improved robustness to clutter. Compared to a baseline operating on unsegmented images, recognition error improves from 8.1% to 4.4% ($n = 1,584$), and the precision of concept detection from 31% to 41% (MAP, $n = 4,160$).

Altogether, these contributions suggest that web video can form the basis for a novel way of concept learning beyond the manual acquisition of small training sets that constitutes the state of the art. With the technology described in this thesis, we can now build concept detection systems that can learn thousands of concepts and offer a better support for video search.

Contents

List of Figures

List of Tables

This work was supported by the PhD Program of computer science at the University of Kaiserslautern, by the German Research Foundation (DFG), project MOONVID (BR 2517/1-1), and by the EU Safer Internet Programme, project FIVES (SIP-2008-TP-131801).

Chapter 1

Introduction

Over the last decade, digital image and video content has become an integral part of our everyday life — we capture it using cameras and camcorders, we store it on local hard-drives and share it with friends, we upload it to the internet and view it on demand. We also use images and video as a source of information, we debate about it, or we are simply amused and entertained by it.

Correspondingly, more visual content is being produced, published, and consumed than ever before in history: web-based image and video sharing portals like Flickr[1] or YouTube[2] are known to millions of people, who upload $65,000$ video clips [YOU06] and $2.5 - 3$ mio. pictures [Auc07] each day. It has been estimated that digital video will account for 91% of all internet traffic in 2013 [Inc09]. In parallel, Companies like Google use imaging sensors to index the world's documents[3], maps[4], or cityscapes[5]. Millions of surveillance cameras monitor our everyday life [BBC06], and large-scale digitization efforts produce video archives containing decades of TV and radio broadcast [Hig06, SOU09]. All these examples show that digital images and video have arguably become an essential source of information and entertainment to a wide community of users.

To employ all this content to its full potential, users must be supported with an efficient search. For this purpose, several strategies have been proposed, like "query-by-image" [M. 95], where the user provides a picture and the system returns

[1] www.flickr.com
[2] www.youtube.com
[3] http://books.google.com/
[4] http://maps.google.com/
[5] http://maps.google.com/help/maps/streetview/index_.html

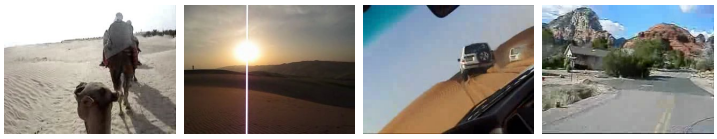
visually similar content, or "query-by-text", where the user enters a few keywords and retrieves content that is linked with these terms. The focus of this thesis will be on the latter approach, i.e. on text-based retrieval. This can be considered standard practice and is realized by services like YouTube or Flickr. However, it requires an *indexing* that links the images and videos in a database with descriptive keywords (or *tags*). The challenge of creating such an index has been referred to as the *semantic gap* [SWSJ00], the discrepancy between low-level content in form of raw pixel values on the one hand and a viewer's high-level interpretation on the other.

Current strategies towards bridging this gap perform an indexing on meta-data like the filename (such as search engines like Google or Yahoo!) or rely on user-generated tags and descriptions (as in case of Flickr and YouTube). Both these strategies are fairly limited: meta-data like descriptive filenames or surrounding text may simply not be at hand. A manual annotation comes with considerable effort and is at the same time limited in other ways: first, users tend to use only the first few words that come to their mind, i.e., tags are incomplete. Second, users give descriptions with respect to their personal prior knowledge and expectations of the clip, i.e. tags are subjective. Third, users often simply assign the same words to complete image groups or to a video clip of several minutes length. These descriptions do not tell us where exactly specific concepts appear, i.e. they are inherently coarse. To some extent, these problems might be overcome by a more careful manual indexing. This however, would be associated with considerably more effort and is thus simply infeasible in most situations.

This opens the question whether computer systems can automatically link low-level content with high-level semantics. Over the last few years, a breakthrough of such content-based image and video understanding technology has taken place, and some solutions have already been integrated with commercial products: Google image search uses a content-based detection of faces, clip art, and line drawings [GOO09], Picasa employs automatic face recognition for photo albums [BBC08], and video search engines like Blinkx benefit from speech transcript and visual features [BLI09]. Other applications where content poses a useful information source are the detection of copyright infringement [ANV09] or object-based retrieval [KOO09, VIP09].

Concept Detection

All these techniques have made their way from research into commercial products over the last years by solving a specific recognition task, like face detection



(a) (b) (c) (d)

Figure 1.1: These sample images illustrate for the concept "desert" that concept detection poses a difficult challenge to visual recognition systems. Enormous intra-class variation occurs due to changes in illumination (a,b), occlusion and perspective changes (c), and variation in the concept itself (d) (pictures from YouTube).

or the search for visually similar content. This opens the question whether visual recognition can be applied to a wider range of generic concepts, including objects ("airplane") as well as locations ("desert"), programme types ("weather report"), or actions taking place ("interview"). This challenge has been referred as *automatic tagging* [DJLW07], *image / video annotation* [FML04], *high-level feature extraction* [KO07], or *concept detection* [YH08]. Throughout this thesis, the term *concept detection* will be adopted.

Given an input picture or video clip, concept detection systems use statistical models over low-level features derived from its content to compute *scores*, which indicate the probability for target concepts to appear. Thereby, the number of target concepts is usually high (i.e., in the range of hundreds / thousands). This renders the development of specialized techniques (as it has been done for faces [ZCPR03, YKA02], for example) infeasible, and a generic approach is required. Also, while other visual recognition systems are applied in restricted environments, concept detection systems should work on a wide range of domains, including photos [RvBKB08], consumer video [CEJ+07], TV [SOK06], and web video [ZMZP08]. This poses strong robustness requirements: first, systems must cope with well-studied phenomena in computer vision, like changing camera parameters, illumination variations, clutter, or occlusion. Further, other factors like coding quality or high intra-class variation of concepts must be taken into account. Finally, in some cases the presence of a concept may not even be well-defined but prone to subjectivity. These aspects (which are also illustrated in Figure 1.1) render concept detection an extraordinarily difficult challenge, and the annotation quality achieved by state-of-the-art systems is far from the accuracy of a careful manual annotation.

Despite these difficulties, concept detection is of practical interest, as a mapping of content to semantics — even if it is unreliable — can help to solve a variety of tasks. The most important one is image and video search, which is a vital problem [VID08] as data load has become massive. In this context, concept detection is applied by first mining image and video databases for a vocabulary of target concepts. At retrieval time, textual queries made by the user (like "lake") are mapped to appropriate concepts in the vocabulary (which might be "water", "river", etc.), and detection results from the corresponding concepts are aggregated. This way, a text-based search can be realized free of manual indexing on the test data, which is why concept detection has been attributed potential to become an integral part of video search technology [SOK06, SWdR+08]. Other applications include content management tasks, like video feed filtering, content-based recommendation systems [YMH+07], the personal delivery of image and video digest, and context-sensitive multimedia advertising [MHYL07]. Further, concept detection can support users with tagging their content (for example, by suggesting keywords) [SvZ08]. Finally, the detection and blocking of specific concepts (like pornography or violence) is of interest [DPN08, GY08].

We conclude that if we could apply concept detection in large-scale practical applications, we could significantly improve the accessability of image and video data in two ways. First, more efficient search could be granted, as videos could now be indexed on shot level: imagine a 10-minute YouTube clip showing different sights of Paris — while a manual annotation of all scenes is time-consuming, concept detection can achieve this automatically and find specific views of, say, Notre Dame Cathedral. Second, we could access significantly more content, simply because a manual indexing becomes impractical at a certain scale but is still feasible if done automatically. Considering the size and — more importantly — the growth rate of image and video collections [Inc09, Auc07], this is of vital importance.

Unfortunately, a practical large-scale application of concept detection is limited by a fundamental *scalability problem*: concept detection systems are usually based on supervised machine learning techniques [CHL+07, Sno07, WLL+07, YH08] (for an introduction to the field, please refer to Chapter 2), and these techniques require training examples for any concept to be learned. Since target concepts can be visually complex, each one might require hundreds of sample views. So far, this problem has been overcome to some extent by acquiring ground truth labels in joint efforts of the research community [AQ08, NST+06]. Yet, such a time-consuming acquisition restricts concept detection in several ways: first, it limits the number of concepts that can be learned, such that the size of current detector vocabularies is a magnitude below the quantities that are probably needed for an accurate video

search [HYL07]. Second, it has been pointed out that detectors overfit to small manually acquired training sets and generalize poorly [YH08]. Third, keeping track of dynamic changes of users' information needs is infeasible as new concepts of interest emerge (such as "Barack Obama" or "Olympics 2008"). From this, I conclude that concept detection suffers from a lack of proper training data and strategies.

Goal and Outline of this Thesis

To realize a broader applicability of concept detection at a larger scale, the goal of this thesis is to reduce the annotation effort associated with concept learning and thus achieve a better scalability and flexibility of concept detection. Thereby, the focus is on the visual aspects of the problem (which does not pose a strong limitation, since a combination with other modalities like text and audio is usually done in subsequent fusion steps).

To achieve this goal, an approach is presented that employs a variety of novel information sources. These can be used to substitute conventional training data (such that annotation effort for the user is reduced) or to complement it (such that concept detection is improved at no — or negligible — extra cost). Such information can be found at different levels of abstraction in the video stream: on a low level, motion information can give valuable clues for certain concepts. On a medium level, the hierarchical structure of video content as a composition of shots, scenes, or shows can be exploited. Finally, on a high level, novel sources of training data can be investigated. As a successful and scalable concept detection should ultimately exploit all information available, the work presented in this thesis will cover all these levels. While the focus will generally be on aspects that are characteristic to video content (like motion information), some of the presented techniques and results apply to images as well. This will be pointed out on a per-case basis.

More precisely, four strategies are presented for driving concept detection towards less supervision:

1. The use of **web video** as a novel source of training data
2. The adaptation of concept learning to noisy training labels using **relevance filtering**
3. The use of context information by combining concept detection with **style modeling**

4. The use of **motion-based segmentation** for an improved recognition of objects.

Each approach will be addressed in one chapter of this thesis, and will previously be outlined in one of the following subsections.

1.1 Employing Web Video

A first approach to reducing the manual annotation effort associated with concept learning is to investigate alternative sources of training data. For this purpose, Chapter 3 of this thesis proposes web video portals like YouTube, MSN Soapbox, Myspace, etc, from which large quantities of video content can be obtained automatically together with descriptive user-generated *tags*. By employing these tags as class labels, web data can complement manually annotated training sets or even substitute them completely, such that a concept learning free of manual supervision is performed. This offers vital advantages in terms of scalability (more concepts can be learned) and flexibility (adaptation to changing and newly emerging concepts can take place).

On the downside, it is to be expected that web video as a training data source leads to lower detection rates compared to manually annotated training sets. This is due to two reasons: first, web video as a domain is complex, including TV content as well as home video, and potential target domains where concept detectors are applied may differ significantly from web-based training material. Second, the label information associated with web video is coarse and subjective, which can have a severe impact on concept learning.

Chapter 3 investigates whether — despite these problems — visual learning from web video can be successful. A concept detection system named *TubeTagger* is presented that employs content downloaded from the portal YouTube for training. The system integrates several types of visual features, like color, texture, motion information, and a patch-based description. TubeTagger is the first concept detection system learning from web video.

An evaluation is presented in which the system is trained on YouTube content and applied to several target domains, including standard data from the TRECVID benchmark [SOK06]. It will be shown that web-based concept learning can be successful in general, and situations will be pointed out in which web video material should complement or even substitute a manual training.

1.2 Relevance Filtering

One characteristic of web video is that its tags are context-dependent, subjective, and coarse, a phenomenon that will be referred to as *label noise* in the following. This causes problems for concept learning, as significant amounts of training material are not visually related to the target concept. To achieve a higher robustness, Chapter 4 of this thesis proposes to adapt the statistical models underlying concept detection such that label noise is explicitly taken into account. This approach will be referred to as *relevance filtering*. It models the relevance of training content as a latent random variable. During training, this variable is inferred, i.e. non-relevant content is identified and filtered out. In contrast to relevance *feedback* [RL03] (which is targeted at a refinement of retrieval results at query time), relevance filtering is applied at training time and combines the elimination of non-relevant material with concept learning. This can be used as a wrapper around standard supervised models, as is demonstrated for a generative approach (kernel densities) and a discriminative one (Support Vector Machines).

In experiments on web video data, it will be demonstrated that YouTube tags do in fact show significant label noise (typically, only $20 - 50\%$ of material are visually related to the target concept). Also, it is shown that the performance of standard concept detection models based on supervised learning degrades severely when trained on such data. In contrast to this, relevance filtering extensions show a higher robustness to label noise, which is achieved by reliably identifying non-relevant content.

1.3 Style Modeling

A third strategy is based on the observation that pictures and video tend to come in a context: while current standard approaches apply concept detection individually on image or shot level, users tend to produce groups of visually correlated material in practice. Examples for this include multiple snapshots taken over the same holiday trip or video streams coming in temporal units such as scenes, movies, or episodes. These image groups share a certain coherence both in terms of visual appearance and the concepts they show. While this context information has been widely neglected so far, a well-founded probabilistically motivated way of employing it is investigated in Chapter 5.

To do so, concept detection is integrated with *style modeling* from the domain of optical character recognition [MB02, SN05]. The approach assumes the pictures or frames in a group to belong to a latent category (or *style*). To learn these

styles, I will again turn towards web portals, which offer training content enriched with category information. Different annotation models are learned from web categories and serve for an accurate style-specific tagging. Test images can be mapped reliably to an adequate style using their context – this way, a more accurate concept detection is achieved by tagging groups of correlated pictures instead of individual ones.

This novel approach is evaluated on the COREL image dataset and real-world photo stock downloaded from Flickr. In these experiments, different styles correspond to locations and travel scenarios, i.e. the annotation of personal holiday snapshots is simulated. It is demonstrated that style modeling helps image annotation to disambiguate and improves the overall tagging performance significantly. Compared to several baselines that label images individually, relative improvements of up to 100% are reported. Also, the proposed approach achieves the best result reported to date on the well-known COREL-5K benchmark (mean per-word precision/recall: 25% / 39%). This is achieved by modeling context, which is not employed by other methods from the literature.

1.4 Motion-based Segmentation

Motion is investigated as a low-level video-specific feature in Chapter 6. Most concept detection systems make only limited use of this information source — it is ignored entirely [CHL+07, WLL+07] or used in form of low-level descriptors [HN07, Sno07, USKB08a]. Instead, in this thesis a concept detection approach will be presented that uses motion in a different way, namely as a source of segmentation information. Such a *motion-based segmentation* in video can separate objects from the background, even in situations where color and texture alone are not sufficient.

However, motion segmentation is error-prone, as fundamental assumptions like pixel constancy or spatial coherence [BA96] are often violated in practice. Therefore, this thesis first makes two contributions targeted at an improved robustness of segmentation. First, a novel approach for the estimation of a parametric global motion (which is usually associated with the background region) is proposed. This method is based on the RAST algorithm from geometric matching [Bre92], which performs a branch-and-bound search of transformation space and thus guarantees a globally optimal solution. Second, an extension of a *direct* motion segmentation approach – which estimates motion and region boundaries in a joint process – is presented, whereas parametric color models are used as additional segmentation clues. In quantitative experiments on a variety of synthetic motion fields and video

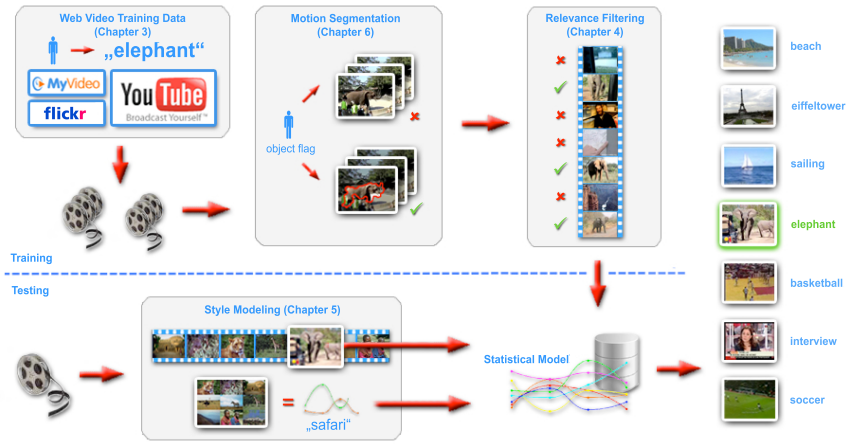


Figure 1.2: An illustration of the proposed concept detection approach applied for the concept "elephant". Web video content is employed for training, non-relevant content is automatically filtered, and optionally motion-based segmentation can be used to separate objects from the background. The resulting model is applied in testing (bottom), whereas a style model achieves an improved annotation by making use of context.

data, both these approaches are demonstrated to give performance improvements over several baselines.

This motion segmentation technology is finally integrated with a patch-based recognition approach, leading to a novel approach for concept detection and object recognition. Motion segmentation serves as a filter, such that only features from the object region are employed for recognition. It is shown that with this approach the robustness of object recognition and concept detection with respect to clutter is significantly improved over the state of the art.

1.5 Framework

Altogether, the aforementioned contributions constitute a novel approach for an efficient and widely unsupervised visual learning from web content. This framework is illustrated in Figure 1.2: to train a target concept (like "elephant"), the system acquires image and video data from web portals like YouTube or Flickr. This can be done fully automatically, or using a refined user query to guarantee a better quality of material. A web-based training set is obtained and refined using relevance filtering, which identifies non-relevant content and relabels it. Addition-

ally, a user can set a flag indicating that a concept corresponds to an object, in which case motion segmentation can be used to achieve an increasing robustness to clutter. Finally, different versions of the framework can be instantiated for different categories (which are associated with different visual styles). Given a video shot to be annotated, context from the same video is used to select an appropriate style (like "Safari"), and a style-specific model is used for concept detection.

Chapter 2

An Overview of Concept Detection in Video

Currently, most commercial video search technology relies on manually generated descriptions and tags. The problem with such an indexing is that it is incomplete, subjective, or just not at hand in many practical situations. Also, its creation becomes increasingly difficult as more and more content is accumulated [Jun09, Sme07] and a manual annotation becomes infeasible. This leads to the question whether computer systems can create indices automatically by inferring the presence of high-level concepts like objects ("cat"), locations ("beach"), or activities ("dancing"), from the content of an image or video. This challenge is the focus of this thesis, and has also been subject to intensive research in content-based video retrieval (CBVR) over the last years. It has been referred to as *concept detection* [NS04], *high-level feature extraction* [OAKS07], *automatic video indexing* [SW05b, WLL+07], or (for the domain of still images) *annotation* [LLM03] or *tagging* [JM04].

In the following, a compact and general overview of research in the area will be given, with the focus on the video domain. Other approaches that are more specifically related to the contributions made in this thesis (including some related approaches for still images) will be covered in appropriate chapters later. I will start with a definition of concept detection (Section 2.1), will then address the most important application areas in Section 2.2, will survey the most frequently used methods in Section 2.3, and will finally discuss issues related to the manual supervision required for concept learning in Section 2.4.

2.1 Problem Statement

The purpose of concept detection is to analyze the audio-visual content of a video and automatically infer keywords (or *tags*) indicating the presence of semantic concepts. So far, we have only stated that these concepts can be associated with a wide variety of semantics, including objects as well as scene types or activities. To understand the notion of a "concept" in more detail, let us refer to the term *relevance* in information retrieval: we assume that a concept is *matched* by certain videos or images that are *relevant* to it. This means that a concept is ultimately defined by all its relevant content.

For standardization purposes, concepts have also been described by textual definitions, like t_1 := "one or more people playing soccer" (LSCOM Concept 017, "soccer" [LSC]). It is important to note that such a textual description is not sufficient to fully explain a concept. For example, consider the following alternative definitions:

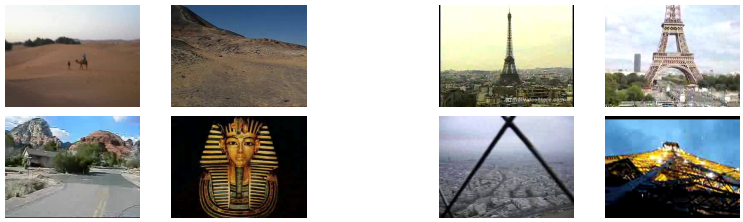
t_2: "shots about soccer"

t_3: "shots that User A would label with soccer"

t_4: "shots that a YouTube user has labeled with soccer"

Obviously, these concepts are similar to t_1 but not identical. First, when comparing t_1 and t_2, it becomes clear that concepts need not necessarily refer to a visibility of certain objects (as t_1), but other, subtle relationships may define relevance (for example, a soccer shoe advertisement or an interview with soccer legend Pelé may be relevant for t_2). Looking at t_3, things get even more complicated: the view of a user may differ from what others consider soccer. Even t_3 itself — i.e., the view of a single user — is not well-defined, as A's expectations and background knowledge may change over time (and with it his understanding of a concept).

An illustration of these problems is given in Figure 2.1 for the concepts "desert" and "eiffeltower". In some cases (top row), most users would agree that the concept is present (for example, see the "desert" shots showing sand dunes). In other cases, different users might have different opinions (for example, a landscape might be considered "desert" or "prairie", or the relation between "desert" and the ancient Egyptian culture might be taken into account or not). Even for seemingly well-defined concepts such as objects (like "scenes that show the Eiffel Tower"), ambiguity occurs: the object may be difficult to recognize, and a user's assessment whether the concept is visually present may be different after he/she has visited Paris and climbed the tower.



(a) (b)

Figure 2.1: Borderline cases of concept presence are frequent, as illustrated here for the concepts "desert" (a) and "eiffeltower" (b). In contrast to obvious cases (top row), in many situations concept presence is difficult to judge (bottom), as concepts are difficult to recognize or prior knowledge is required (pictures from YouTube).

To some extent, this problem can be overcome by a precise textual description of what is *meant* with the concept, as it has been done in concept detection research [NST+06]. An alternative for choosing less ambiguous concepts is indicated in t_4, whose definition is derived from real-world data, and user subjectivity is addressed by relating concept presence to the tagging behavior of the whole YouTube community.

Overall, despite the aforementioned difficulties, it is usually assumed in concept detection that concept presence is well-defined, and we will adopt this view throughout this thesis. Yet, it should be kept in mind that this assumption is not correct, and that ambiguity is inherent to practical concept definitions.

Definition "Concept Detection" Given the fact that a concept t is ultimately defined by its relevant content, concept detection is the problem of deciding whether a video X is relevant for t. The input of a concept detection system consists of (1) the video X, which can be a long clip or a single shot (i.e. a scene not interrupted by any cuts), and (2) a vocabulary of target concepts (or *tags*) $t_1, ..., t_n$ as defined above. The goal of concept detection is to estimate **scores** $\phi_{t_1}(X), ..., \phi_{t_n}(X)$ indicating whether each concept appears in X. These scores may be interpretable as probabilities, but this is not necessarily the case — it may be sufficient that videos can be ranked by sorting them according to their score. It is important to note that this multi-class problem is usually divided into many binary classification problems: the score for each concept is estimated separately, and correlations between concepts are taken into account in subsequent postprocessing steps. This approach will be followed throughout this thesis.

Usually, the mapping $\phi_t(.)$ is modeled using a statistical classification algorithm (e.g., a neural network or a Support Vector Machine). Previous to concept detection, the parameters of this classifier $\phi_t(.)$ are learned in a training step. For this purpose, training videos $x_1, ..., x_n$ are given, with labels $y_1, ..., y_n \in \{-1, 1\}$ denoting the presence/absence of t.

2.2 Applications

One fundamental characteristic of concept detection is its generality: concepts can cover a wide range of semantics, including objects, scenes, activities, etc. On the one hand, this renders the task a challenging problem with high intra-class variance and hundreds or thousands of categories involved, and (depending on the concept) classification accuracy can be low. On the other hand, this generality makes concept detection applicable in a variety of practical use cases, where a linking of low-level content with high-level semantic descriptions (even if it is inaccurate) is of interest:

- **Video Search**: The most prominent application of concept detection is the text-based search in video databases. For each concept in a predefined vocabulary, a detector returns shots from the database that are most likely relevant for the concept. Textual queries formulated by users are then mapped to the concept vocabulary, which can happen manually [CH05], by using concepts as filters [Sme07], or employing an ontology [SHH+07]. Alternatively, concepts can define a *semantic space* [TNS07] and form the basis of further machine learning techniques.

  While early studies on the utility of concept detection for retrieval were rather sceptical [CH05], rapid improvements could be made in terms of mapping techniques and vocabulary size. Correspondingly, concept detection (though not as accurate as a careful manual annotation) is now attributed potential to become a key building block in modern content-based multimedia search systems [HYL07, Sme05, SWdR+08], and several CBVR systems exploit lexicons of visual concept detectors [SWdR+08].

- **Video Content Management**: While video search is the main focus of concept detection, several related use cases exist in a broader video management context. These include the content-based personal delivery or recommendation of video digest [YMH+07], or context-sensitive multimedia advertising [MHYL07].

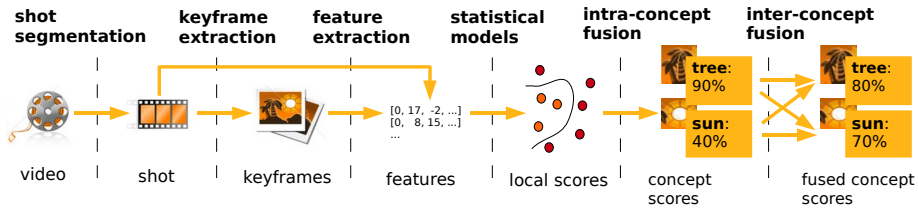


Figure 2.2: The processing pipeline of a typical concept detection system. An input video is segmented into shots, from which representative keyframes are extracted. Shots and keyframes are described by numerical features, which are fed to statistical models estimating concept scores. These scores are fused over keyframes and over different feature modalities (*intra-concept fusion*), and are finally refined using correlations between concepts (*inter-concept fusion*).

- **Video Annotation**: Another application of concept detection is the annotation of video content. Obviously, tags can be derived directly by thresholding concept scores. Even though the performance of current detectors may not be sufficient for such a fully automatic keyword assignment, systems can support users with tagging their videos in a semi-automatic fashion [SvZ08] (for example, by suggesting keywords).

- **Specific Concepts**: Finally, the detection and blocking of specific concepts (like pornography or violence) is of interest to search engines or law enforcement [DPN08, RJB06, GY08].

2.3 Methods

Soon after the development of first practical concept detection systems in the early 2000s [NH01], the TRECVID campaign [Sme05] was established, a video retrieval benchmark providing researchers with uniform evaluation procedures and standardized datasets. Research efforts on concept detection focus in TRECVID (with over 40 research groups participating in the "High-level Features" task [KO08]), though aside from TRECVID itself other efforts towards standardization and comparability are being made by sharing intermediate results like features, annotations, and trained detectors [SWvG$^+$06, YCKH07].

Figure 2.2 illustrates the processing pipeline of concept detection as it has been introduced in [ABC$^+$03] and is followed by the majority of systems. Video search

is usually done on shot level, though other possibilities have been pointed out to be of practical interest [Sme07] (for example, in a video annotation scenario, concept detection should operate on video clip level). An input video is first segmented into shots, i.e. scenes captured by a single camera without any interruptions by cuts or similar transitions. Afterwards, features describing the content of each shot are extracted. This can be done by selecting representative *keyframes* and applying an image-based feature extraction, or by directly extracting features like motion patterns from the video stream.

After this, features are fed to a statistical modeling, which estimates *scores* indicating the presence of target concepts. Several models and feature types have been proposed, each giving a different score. These are fused for each concept (*intra-concept fusion*). Finally, correlations between concepts are taken into account in an *inter-concept* fusion. These steps are discussed in processing order in the following.

2.3.1 Shot Segmentation

Shot segmentation (or *shot boundary detection*) is targeted at an automatic temporal segmentation of the video stream. For this purpose, shot transitions need to be detected, which can be hard cuts, fades, wipes, or similar effects. Usually, shot boundaries are detected as sudden changes of visual appearance within pairs (or sequences) of subsequent frames. Several surveys give overviews of popular features and decision rules [Han02, Lie01, YWX+07]. The problem is well understood, and (particularly for the most frequent hard cuts transitions) a high accuracy can be achieved.

2.3.2 Keyframe Extraction

The goal of keyframe extraction is to select frames that represent the visual content of a shot well. This step is motivated by two reasons: first, computational cost is reduced significantly compared to a full usage of all frames in a shot, and second, video data is reduced to images, for which a basis of well-understood features and statistical models exists.

Often, very simple keyframe extraction methods are used that extract a single keyframe per shot, for example the center frame [Sme07, WCGH99]. More elaborate techniques have been presented for *video summarization*, where the selection of proper keyframes is critical for visualization purposes. These methods use adaptive techniques to extract a number of keyframes per shot. One cate-

gory of approaches estimates candidates for keyframes at points of strong content change [ADDK99]. Others use unsupervised learning techniques and return cluster representatives as keyframes [HM00, HZ99, MRY06].

In the context of concept detection, such adaptive methods are not in the main focus of research. However, it has been demonstrated before that increasing the number of keyframes improves concept detection [Yua07, SWG+05, USKB08a].

2.3.3 Feature Extraction

A variety of features has been investigated in concept detection to describe the content of video frames and serve as discriminative input for statistical modeling. A full survey of descriptors is far beyond the scope of this thesis (please refer to overview papers like [DKN08, SWSJ00]). Instead, in the following a characterization of the most frequently used types of features is given. While the majority of approaches is based on earlier work for still images, additional video-specific features will be covered as well, like audio, motion, or overlayed text.

Color Color — a standard feature in image retrieval — is also frequently used in concept detection. Popular examples are color histograms [WLL+07] or moments of the RGB color channels [YHC07, YH08].

To include information on the spatial layout of color, features are usually binned by partitioning frames into grids (or by segmenting them automatically), and features for the single partitions or regions are concatenated. This principle applies to all visual feature categories in the following.

Texture Properties of image texture like coarseness and orientation can be discriminative cues for concept detection. They are represented using histograms over Tamura features [TMY78], or by filtering with banks of Gabor features at varying orientation and scale (followed by an aggregation of results via moments) [YHC07, YH08]. As alternative filters, Haar wavelets have been proposed [CHL+07].

Edges Edges make features that are strongly related to image texture. Descriptors of this category are mostly histograms over the orientation of edges [WLL+07], which can be extracted using a standard detector (e.g. Canny's method [Can86]). A soft version of this feature based on the image gradient is called *histogram of oriented gradients* (HOG) [DT05], and has also been successfully applied to concept detection [CHL+07].

(a) (b) (c)

Figure 2.3: An illustration of a patch-based approach (images from [SREZ05]): given training images (a,b) showing instances of an object category (here, faces), a patch-based approach learns the most discriminative local patches (c) for an object class (here, eye and forehead).

Patches Over the last years, *patch-based* descriptions have become popular in computer vision research, which replace global image-level features (like color histograms) with representations based on collections of local image parts. With this, a higher robustness can be achieved with respect to deformations, clutter, and partial occlusions. An illustration of a typical approach is given in Figure 2.3, where local patches associated with the concept "face" are displayed. These patches are learned from unsegmented training images despite significant background clutter.

This development is based on novel feature detectors and descriptors with strong invariance and robustness properties, which allow to detect and match similar features even in case of strong changes of scale, illumination, and viewpoint [BTvG06, KB01, Lin98, Low04, MCMP02, MS04]. Overviews and quantitative evaluations of local features can be found in the literature [Rot08, Mik03, SMB00]. Recognition systems that are based on these methods have been studied intensively in computer vision research over the last decade [BWP00, DKN05, FFP05, FML04, FPZ03, HL04, MLS06, SZ03, UVNS02].

Outstanding popularity has been achieved by a data-driven discretization of visual features in a clustering process, which is usually referred to as the *bag-of-visual-words* (or *bag-of-features*) model [FFP05, QMO$^+$07, SREZ05, ZMLS07]. This approach draws an analogy to the well-known *bag-of-words* representation from text processing [Lew98]: just like a document is represented by counts of occurrences of words, a visual document (or image, respectively) is represented via counts of *visual* words, i.e. categories of image patches. These categories are typically estimated using a clustering (an alternative is a supervised training on patches manually drawn from segmented image regions, as for van Gemert et al.'s *protoconcepts* [vGV$^+$06]).

Models based on the resulting descriptors have demonstrated an excellent performance in object category recognition benchmarks [EZWvG06], and have also recently been applied successfully to concept detection in video [JNY07, SZB08, vdSGS08].

Motion While all features above are extracted from static keyframes, some previous work exists that employs the dynamic content of video in form of *motion* features. These are usually extracted from 2-dimensional motion vectors in the image plane, which can be obtained using a tracking of sparse features [TK91] or a dense estimation of optical flow [BB96]. For video content, such motion fields are also directly encoded in the video stream, which can form the basis of fast, compressed-domain features like motion histograms [ACAB99, MZ03]. Such features have also been demonstrated to improve concept detection compared to static image descriptions [HN07, USKB07, USKB08a, Sno07].

A related idea of capturing dynamic video content is to extend patches [SMB00] from static images to *spatio-temporal* ones extracted from the video volume (the time domain is included as a third dimension). Interest point detectors are used which are attracted by points in the video volume showing both salient image features and motion changes. These have been studied in the context of human action recognition, and Schindler et al. have also tested them for web video tagging [SZB08]. Despite these efforts, the concept of motion has been studied to a limited extent and is not used to its full potential yet. Its further investigation has been called a key challenge of content-based video retrieval [Sme07].

Text Another valuable clue is text information in video. It appears in the scene itself (for example on road signs or T-shirts), it is overlayed during post-production, it is added as meta-data in form of closed captions, or it appears as spoken language in the audio track. In all these forms, text provides a strong feature for concept detection, even if it is degraded due to weaknesses of speech recognition or OCR [WCGH99].

The extraction of scene text is currently an active research area of computer vision [Luc05] and has not reached the maturity of being applicable to concept detection. Key burdens are compression artifacts and strong variation of text scale and design. In contrast to this, overlayed text can be successfully extracted at reasonable recognition rates using optical character recognition (OCR) [WCGH99]. Spoken language can be extracted to some extent using automatic speech recognition (ASR), which is also used as a feature in TRECVID [HOdJ07].

2.3.4 Statistical Models

We use statistical models to make a decision of concept presence based on a video's low-level features. Given an input feature vector associated with a keyframe or shot, a statistical model estimates a numerical score ϕ_t indicating the presence of a target concept t. In a probabilistic setting, these scores can be interpreted as a posterior of concept presence, but this is not necessarily the case, as scores are often only used for a ranking of items. Usually, the estimation of ϕ_t is treated as a classification problem with two classes, concept presence and absence. The model ϕ_t is learned from a set of samples $x_1, ..., x_n \in \mathbb{R}^d$ with labels $y_1, ..., y_n \in \{-1, 1\}$ indicating concept presence. A variety of models has been suggested in the pattern recognition literature [DHS00]. Since a full survey is far beyond the scope of this thesis, only the approaches most frequently used in concept detection are briefly outlined in the following.

- **Support Vector Machines (SVMs)**: Support Vector Machines are one of the most widely used classification algorithms today. They are also a popular choice in concept detection, which is empirically motivated by excellent results obtained in standard benchmarks [KO08].

  SVMs are based on two fundamental ideas. The first one is linear maximum-margin classification, i.e. the decision boundary separating classes is chosen to be a hyperplane maximizing the distance from the training samples x_i. The second idea addresses the fact that in many practical situations non-linear decision boundaries are required. This is achieved by mapping samples x_i to a potentially high-dimensional space $\mathcal{H}$ using a function $\Phi : \mathcal{R}^d \rightarrow \mathcal{H}$. As only the computation of the inner product $K(x, y) := \langle \Phi(x), \Phi(y) \rangle$ is required, we can abstract from the space $\mathcal{H}$ and only compute the similarity (or *kernel*) K, which has been referred to as the *kernel trick*. An in-depth introduction to SVMs and their theoretical properties can be found in tutorials [Bur98] and in the literature [SS01].

- **Maximum Entropy**: Like SVMs, Maximum Entropy follows the idea of discriminative classification, i.e. the decision boundary between classes is modeled directly. However, while SVMs choose this decision boundary by margin maximization, Maximum Entropy follows a different strategy: training data is used to impose *constraints* on the class posterior $P(c|x)$, but $P(c|x)$ is chosen to be as uninformative as possible otherwise. This can lead

to a posterior of the form:

$$P(c|x) = \frac{1}{Z} \exp\left(\lambda_0 + \sum_i \lambda_i \cdot x_i\right).$$

where x is the input sample, c the class, and λ a parameter vector. An introduction to the approach can be found in [NLM99]. The method has been applied to concept detection [ABC+03] and image annotation [JM04].

- **Nearest Neighbor (NN)**: nearest neighbor matching [DHS00] offers a simple, transparent, and intuitive approach: a sample is classified by finding the most "similar" training samples and adopting their class labels. More precisely, if finding K such nearest neighbors $x'_1, ..., x'_K$ with labels $y'_1, ..., y'_K$, the posterior for class c is set to:

$$P(c|x) \approx \frac{|\{x'_j | y'_j = c\}|}{K}.$$

A number of other models has been employed for the annotation of still images, but has not (or only marginally) been investigated for video to the best of the author's knowledge. Examples include generative mixture models [CCMV07], topic models [FFP05, MGP04], or relevance models [FML04, LLM03].

2.3.5 Intra-Concept Fusion

Different keyframes of a video, features associated with them, and statistical models provide different concept scores. To combine this information to a global score, two general strategies exist: First, *early fusion*, where different features are concatenated before classification. While this strategy offers the benefit that all information is available to the classifier simultaneously, the combined feature vectors can be high-dimensional, such that the resulting methods are often inefficient and prone to overfitting. *Late fusion* offers a simple alternative by applying keyframe- or feature-wise classifiers and combining their *scores*. To do so, several strategies have been suggested:

- Simple **heuristic schemes** that set the fused score to the maximum, minimum, median, product, or mean of the input scores. Such combinations are simple and fast to compute. They can be motivated by a probabilistic interpretation under score independence (for the product rule [LH02]), or by a high robustness to incorrect outlier scores (for the sum rule [KHDM98]).

- **Re-ranking methods**, which do not operate on scores but fuse several ranked retrieval lists to a final output list. This fusion can be done by minimizing an average distance with respect to the input lists, or by treating rank as a score as by Borda's method and variants [vES00].

- **Classifier combination**: more generally, intra-concept fusion can be seen as a combination of classifiers, for which we can refer to a variety of well-known techniques. For example, Lin and Hauptmann apply standard classification methods to the input scores [LH02]. Other possibilities are stacking, cascading, or boosting [DHS00, Ch. 9]. The benefit of these techniques is that — since they are supervised and employ class labels — irrelevant features can be identified and given lower influence on the final score.

2.3.6 Inter-Concept Fusion

So far, we have addressed the design of an independent detector for each concept. In practice, however, the occurrence of tags can be strongly correlated: for example, the presence of the concept "car" is heavily related to the presence of "outdoor" and "street".

It seems reasonable that concept detection should take this information into account. As a joint detection of concepts is infeasible due to combinatorial problems, inter-concept correlation is usually modeled in an additional postprocessing step referred to as *inter-concept fusion* [WLL$^+$07]. This can be seen as another classification problem, where concept-wise scores serve as input features and an overall score is to be computed. For this purpose, neural networks have been tested [DZ07], and Jiang et al. [JCL07] propose a probabilistic formulation based on conditional random fields with potentials over concept pairs.

Overall, it has been demonstrated that concept detection can be improved significantly using a context-based fusion step. Correspondingly — though it is not the focus of this thesis and will be omitted in the following — it should be kept in mind that inter-concept fusion could still be added as a post-processing step.

2.4 Levels of Supervision

So far, the setup and internal structure of concept detection systems have been characterized. Particularly, the use of adequate statistical models over content-based features has been pointed out to be a key component. All statistical mod-

els outlined so far require training frames $x_1, ..., x_n$ with associated class labels $y, ..., y_n \in \{-1, 1\}$ indicating concept presence. This approach is referred to as *supervised learning*.

Note that — as the number of target concepts is large — the effort associated with acquiring labeled training data is enormous. Therefore, the key question addressed in this thesis is whether we can acquire alternative information sources such that concept detector training can be performed with lower annotation effort. This alternative information might come in form of more samples x_i and labels y_i. In this case, we remain in the standard supervised learning setup. However, other kinds of information might be of interest as well:

- **Presence / Absence of Image Segmentation**: Often, a concept in an image or frame is not related to the whole picture, but only to a certain region in it (like "faces" in a portrait picture). The rest of the image is background (or *clutter*) which is weakly related to the concept or not at all. Note that the standard setup mentioned above does not provide us with the information *where* in the image the concept appears, i.e. concept learning must be robust with respect to clutter.

  Alternatively, if training images came with additional segmentation information, the learning of concept models could be simplified (as clutter has no influence) and better concept detectors could be expected. For example, Snoek et al. [SWG+06] train a small set of generic concepts on segmented images.

- **Presence / Absence of Temporal Segmentation**: An analogy to image segmentation can be drawn for the temporal dimension of video. For training, a concept detection system is usually given a set of frames with labels indicating for all of them whether the concept appears or not. While this information can be difficult do provide, alternatively, we might give the system long video clips and only tell it *whether* the target concept appears *at some time* in a video, but not *when exactly*. Compared to the supervised scenario above (where each frame needs to be labeled), this setup would require significantly less annotation effort.

Note that both these definitions take additional structure of video content into account, namely the fact that frames are composed of pixels, and the fact that frames come in temporal sequences. This information cannot be modeled using a plain supervised learning setup with frames as samples and frame-level labels.

Therefore, one focus of this thesis is to drive concept detection towards less supervision by employing information beyond keyframe level. Thereby, the understanding of "supervision" is a practical one: when referring to *weakly supervised* concept detection, I mean that less manual annotation effort is involved in system training or that concept detection is improved at no extra cost. This does not necessarily refer to the fact that fewer labels are provided — it can simply mean that label information is acquired from other sources (Chapter 3), that labels are coarser (Chapter 4), or that additional segmentation information is inferred automatically (Chapter 6).

Chapter 3

Concept Learning from Web Video

The effort associated with the manual acquisition of training examples poses a key challenge to concept detection. To overcome this problem, this chapter suggests *web video* as a novel source of training data, offering a scalable and flexible concept learning. The main contributions of this chapter are[1]:

1. A system is presented that learns to detect concepts by automatically downloading training material from the video sharing web-site YouTube. This system performs an autonomous learning, which can scale to thousands of target concepts and keep track of dynamic changes.

2. It is demonstrated that YouTube-based detectors generalize comparably well to novel target domains as detectors trained on manually acquired training sets (a moderate relative performance loss of 11.4% occurs [$n = 917,662$]).

3. It is shown that YouTube content can complement manually acquired training sets and improve generalization capabilities (relative performance improvement 11.7% [$n = 917,662$]).

From these results, I draw the conclusion that web video cannot only complement manually acquired training data, but can replace it entirely when generalizing to new domains. This way, a detection accuracy comparable to the state of the art can be preserved, and manual annotation effort is overcome.

[1] This chapter is based on the author's work in [UKSB08, USKB07, USKB08a, USKB09]

3.1 Introduction

Several applications in video retrieval — like search or recommendation — are based on textual representations indicating the presence of semantic concepts, like objects, persons, locations, and activities. In many practical situations, such a textual description is not at hand, and a complete manual labeling is infeasible due to the enormous size of today's video databases. To overcome this problem, concept detection (or *video tagging*) systems have been developed that automatically infer the presence of concepts directly from the video content.

While concept detection has been implemented in research prototypes [CHL+07, Yua07, Sno07], it has not been applied in practical large-scale settings yet. A key reason for this is that state-of-the-art systems are based on supervised machine learning techniques, and that these techniques require video content labeled with target concepts for training purposes (an overview of the most frequently used approaches has been given in Chapter 2). Currently, this training information is acquired manually, i.e. operators annotate video data according to precise visual criteria [NST+06]. The quality of the resulting training content is high in the sense that the annotated concepts are carefully selected with respect to usefulness and feasibility of detection, that clear, restrictive definitions of concepts are specified, and that a precise annotation is done on shot level [NST+06].

On the downside, the effort associated with such data acquisition is enormous: first, each target concept may be visually complex and thus require hundreds of training samples. Second, the number of tags to be learned is high (in the range of thousands) [HYL07]. Finally, concept detection systems tend to overfit to training sets and generalize poorly to video content unseen in training, a problem that is even more severe if a switch between different video domains takes place (for example, from news video to home video) [YH08].

While concept detection research has strongly focused on news video so far, other web-based video collections have emerged over the last years, like YouTube[2], Blinkx[3], Myspace[4], and many others. These services allow users world-wide to share all kinds of video, ranging from TV news and documentaries over movie scenes to home user content, like holiday clips or video blogs. Also, they have set the platform for entirely new genres like interactive web-based series [Pat08]. For retrieval purposes, these portals rely on textual descriptions and keywords (*tags*) provided by users during video upload.

[2] http://www.youtube.com
[3] http://www.blinkx.com
[4] http://www.myspace.com

Web video offers a large-scale, publicly available information source enriched with tags and descriptions, which are provided by a community of millions of users. The key idea of this chapter is to employ this information for concept learning. A system is presented that implements this approach and performs a visual learning from YouTube (correspondingly, the prototype is named *TubeTagger*). When triggered to learn a concept, the system downloads videos tagged with the concept and uses them for training. This way, web video can complement other training sets or substitute them entirely, such that an autonomous concept learning free of manual annotation effort takes place.

On the downside, training on web video poses a difficult challenge compared to learning from state-of-the-art datasets specifically acquired for research purposes. This is due to the enormous diversity of web video material, and due to its coarse and unreliable label information. Therefore, the key question addressed in this chapter is whether — using a state-of-the-art concept detection approach — visual learning from web video can be successful. To answer this question, a quantitative evaluation is presented in which the TubeTagger prototype is trained and tested on a variety of video data, including web video as well as standard datasets from the TRECVID benchmark [SOK06]. In these experiments, it will be demonstrated that concept learning from web video is feasible, and that YouTube-based detectors generalize comparably well to different domains as the ones trained on manually acquired data.

The chapter is organized as follows: First, an overview of standard datasets and annotations is given, and benefits and limitations of manual training data acquisition are discussed (Section 3.2). Second, the idea of concept learning from web video is introduced in more detail (Section 3.3). Related work in the context of learning from web data is presented in Section 3.4. The TubeTagger prototype is introduced in Section 3.5, and quantitative experiments are described in Section 3.6. A discussion concludes the chapter (Section 3.7).

3.2 State of the Art

The machine learning techniques underlying concept detection systems require training sets of video data with labels indicating the presence of target concepts. The standard approach is to acquire this information manually. As this is a time-consuming (and thus cost-intensive) process, the research community has established joint annotation efforts on standard datasets [NST+06, KO08, AQ08]. This information is shared for evaluation purposes, which allows a straightforward com-

parability of results and makes it possible for researchers to participate without the overhead of manual labeling. Concept detection has made large steps forward due to this approach, and significant progress on increasingly difficult datasets could be recorded over the last years [KO05, KO06, KO07, KO08].

3.2.1 Datasets

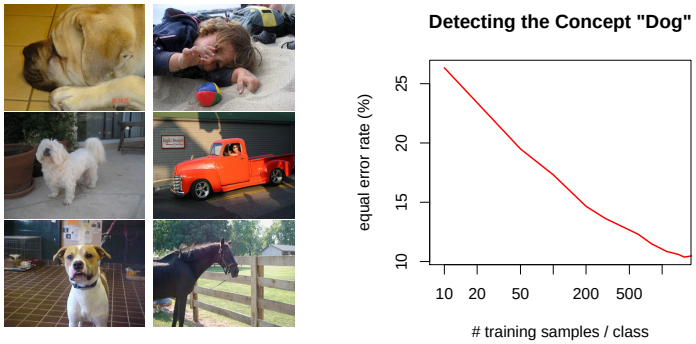
Standard datasets for concept detection have been acquired for evaluation purposes in the TRECVID benchmark [SOK06] or by related efforts from inside the research community. The resulting datasets are of a high quality in the sense that precise shot-level annotations are provided according to relatively clear criteria, and that concepts are selected with respect to a feasible detection. Fore example, operators assign labels on shot-level according to definitions like "shots that take place (outdoors) at night. [...] Excluded are sports events under lights" (concept "nighttime", LSCOM dataset [NST$^+$06]).

Since the focus of TRECVID has long been on news video, concepts are often chosen to be characteristic for this domain [NST$^+$06]. Accordingly, annotations are provided on news video data, which applies for all of the following datasets:

- **TRECVID**: TRECVID's "High-level Feature Extraction" task addresses concept detection as a building block of video retrieval. Each year, participating research groups submit detection results for a small number of concepts (usually $10-20$), which are then manually assessed on video collections of news or documentary TV. The resulting pool of annotations is made publicly available[5].

- **LSCOM**: Concept detection researchers do not only use common video data, but have also designed common vocabularies of semantic concepts to be detected. One such vocabulary called LSCOM (*Large-scale Ontology for Multimedia*) has been created in 2005. It consists of $1,000$ concepts enriched with semantic relations forming a multimedia ontology. Concepts for the lexicon were manually selected by a consortium from research and industry according to the following criteria [NST$^+$06]: (1) *utility* — concepts should support typical real-world retrieval use cases (2) *coverage* — the semantic space of potential user interest should be covered well (3) *feasibility* — an automatic detection of concepts from video content should be possible in general, and (4) *observability* — concepts should occur frequently in standard datasets to

[5] http://www-nlpir.nist.gov/projects/trecvid/trecvid.data.html



(a) (b)

Figure 3.1: Detecting the concept "dog". (a) Sample pictures showing dogs (left) or not (right). Strong intra-class variation can be observed. (b) The equal error plotted against the number of training samples ($n = 4,000$, please note the logarithmic x-scale). Recognition error converges at several hundreds of training samples (pictures taken from petfinder.com and Flickr).

allow for statistically significant benchmarking results. Annotations on the TRECVID'05 video data are available for 449 concepts [LSC].

- **LSCOM-lite**: As an interim result of the LSCOM effort, a smaller test vocabulary called *LSCOM-lite* has been published [NKK+05]. 39 concepts related to news video retrieval were selected in order to cover a diversity of potential user interest. Concepts include program categories ("weather"), scene settings ("outdoor"), objects ("airplane"), and activities ("people running"). Annotations on the TRECVID'05 video data are available [LSC].

- **Mediamill**: Researchers from Amsterdam University have designed a challenge problem for a component-based evaluation of video retrieval called *Mediamill Challenge* [SWvG+06]. The challenge provides a lexicon of 101 concepts obtained by enriching the 39 LSCOM-lite concepts with further sample tags. Detectors, baseline results, and extensive annotations are provided on the TRECVID'05 video collection.

3.2.2 Limitations

The manual acquisition of datasets like LSCOM can be considered state-of-the-art in concept detection research, and joint community effort has lead to concept vocabularies of hundreds of tags. Further, a high comparability of research results has been achieved and driven the field towards an applicability in practical video search scenarios. Yet, concept detection remains strongly limited by the cost and time associated with manual annotation.

This is due to several reasons. First of all, the intra-class variance of many concepts is high. For example, pictures showing the concept "dog" vary significantly with background clutter, object pose, camera perspective, and lighting. Further variation occurs between instances of a concept, like in the case of "dog" between different breeds. The consequences for concept detection are illustrated in a small experiment: a training set of images showing dogs was acquired from the web[6] and classified against non-dog pictures randomly sampled from Flickr. A state-of-the-art concept detection approach was applied (visual word features, SVM classifier — for details, please refer to Section 3.5), and the equal error rate was measured on a held-out test set of $4,000$ images. Sample pictures and results are given in Figure 3.1. It can be seen that the error (averaged over 5 runs of resampling) decreases with a growing number of training samples. For example by increasing the training set size from 100 to 2000, classification error can be reduced by 40%. This indicates that — for state-of-the-art methods and semantic concepts of intermediate complexity — training sets of several hundred positive samples are required. Annotating datasets of this size is a time-consuming task: estimates for labeling a single concepts are in the range of $15 - 45$ hours, as has been reported in TRECVID'08 annotations[7] and confirmed in experiments conducted for this thesis. For the LSCOM effort (where annotations for a vocabulary of 449 concepts were acquired) a cost of $6,000$ man hours has been reported [KHN+06].

Further, the number of concepts required for practical applications like video search is high, as a wide range of potential user queries needs to be covered by concept detectors. For example, Chang et al. [CHJ+06] reported that increasing concept lexicon size from 39 to 374 concepts improves the number of queries to be answered by 50% and the overall retrieval performance by 100%. An outlook on what numbers of concepts might ultimately be required for practical high-quality video search is given by Hauptmann et al. [HYL07]. It lies in the range of $3,000 - 5,000$ concepts, and has been restricted to the domain of news video.

[6] http://www.petfinder.com
[7] http://mrim.imag.fr/tvca/

While for general-purpose video search a significantly higher number of concepts is probably beneficial, current prototypes utilize no more than a few hundred concepts [NST+06, YCKH07] simply because training sets are time-consuming and cost-intensive to acquire.

When taking these facts into account, it is obvious that — though techniques for reducing the annotation effort exist based on *active learning* [AQ07] — an explicit annotation of training datasets is impractical. Even if we could acquire annotations for thousands of concepts, significant drawbacks remain. One problem is that ground truth annotations are always bound to an underlying video dataset. Current concept detection systems are mostly trained on specific news TV programmes and only perform well on this data source. It has been demonstrated that systems tend to learn degenerate nearest neighbor solutions, i.e. they simply memorize shots and strongly overfit to the datasets they are trained on. Correspondingly, the generalization capabilities of concept detectors are severely limited [YH08], even between different news channels. To some extent, this problem can be overcome using *cross-domain* techniques [CJYZ07, YYH07], which adapt classifiers trained on a source dataset using only few annotations on the target domain. These techniques have been tested when switching between different news channels [YYH07] and different genres like news vs. documentary programme [CJYZ07]. Yang et al. [YYH07] and Chang et al. [CJYZ07] report improvements by cross-domain adaptation steps. Yet, generalization capabilities — and with it the utility of manually acquired training data — remains limited.

Another severe problem is that annotations are static, and so are the concept detectors trained on them. In contrast to this, the world's video content and users' information needs are constantly evolving. New concepts of interest pop up, like "9-11", "secondlife", or "Barack Obama", and concept detection systems should adapt accordingly. Keeping track of these changes is infeasible using explicit manual annotations.

3.3 Web Video as Training Data

In the last section, it has been pointed out that state-of-the-art concept learning is performed on small-scale sets of manual annotations. The limitations of this approach with respect to the scalability and flexibility of concept detection have been discussed.

In the following, a different data source for concept detector training is investigated, namely web video. Web video is a rapidly growing market, which has

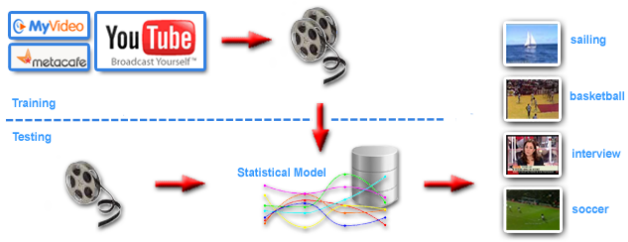


Figure 3.2: Concept learning from web video: a system autonomously downloads a set of training videos from portals like YouTube. From these videos, statistical models for the appearance of semantic concepts are learned, which can then be applied to tag previously unseen videos.

brought up new forms of interactive, highly dynamic video databases linked with textual descriptions and discussions. These portals — YouTube, Blinkx, Myspace, and many others — host content ranging from TV news and documentaries over movie scenes to home videos, like holiday snapshots or video blogs.

From a video retrieval perspective, web video has only been subject to limited study. Yet, it is highly interesting both as an application and as an information source for concept detection. When viewing it as an application, concept detection could offer an improved keyword search, help to group videos into semantic categories[8], or support users with tagging their videos. As an information pool, web video offers a large-scale dynamic source of video data, which is enriched with label information provided by a large community of users.

Surprisingly, web video has not been subject to intensive research so far. Only a few contributions regarding web video as an application exist [SZB08, ZMZP08], and as a source of training data it remains unstudied. To fill this gap, this chapter investigates web video for concept detection. This setup is illustrated in Figure 3.2: when given a target concept to be learned, the concept detection system acquires a collection of training videos from web portals like YouTube. Tags associated with this content can be used as ground truth labels for concept detector training: if a video clip is labeled with the target concept, it is used as a positive training sample, otherwise it is used as a negative one. Based on this information, machine learning techniques build statistical models for the appearance of each target concept. When applying these models to a previously unseen video, scores can be inferred that indicate the presence of target concepts.

[8] http://www.scils.rutgers.edu/conferences/mmchallenge/2009/02/02/google-challenge/

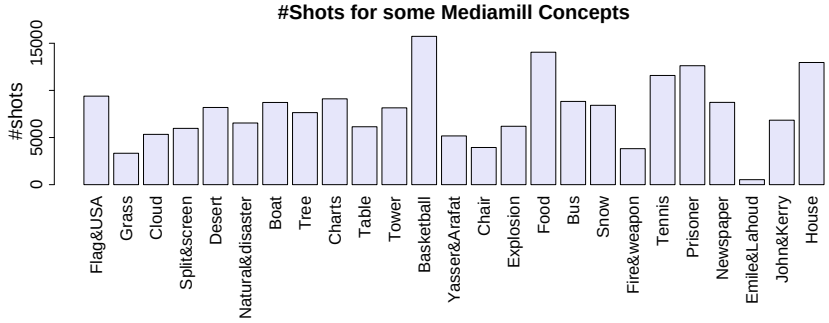


Figure 3.3: Quantity of training material obtained from YouTube for some randomly selected Mediamill concepts [SWvG$^+$06].

The main characteristic of this setup is that an entirely autonomous concept learning is performed, and only minimal manual interference is required. This offers two fundamental benefits:

- **Scalability**: Concept detection can seamlessly scale up to thousands of concepts if enough processing power is available.
- **Flexibility**: Web video portals are highly dynamic, with users uploading 20 hours of video every minute [Jun09]. This content is astonishingly up-to-date: for example, clips of the opening ceremony of the Olympics Games 2008 in Beijing were available a few hours after the event. As web video is constantly updated by its users, the concept detectors trained on it can keep track of new interesting concepts.

To realize such a concept learning, a sufficient quantity and quality of training data must be obtained from web video portals. In the following, a brief discussion of these two issues will be provided. Experiments presented later in this chapter (Section 3.6) will then provide quantitative results of concept detection when training on web video.

3.3.1 Quantity of Training Data

Web video portals offer a tremendous, constantly growing amount of video data. For example, the market leader YouTube hosts 83.4 Mio. videos [YOU], and 65,000 new clips are uploaded each day [USA06]. However, the distribution of concepts

present in this material is highly biased towards popular tags (like "funny", "love", or "girl"), and it is not clear a priori whether enough material can be obtained for the training of certain target concepts.

Therefore, a small experiment was conducted for two standard sets of concepts (20 concepts from the TRECVID'08 benchmark [KO08], and 101 concepts from the Mediamill Challenge [SWvG+06]). Each concept was manually assigned a canonical YouTube category to obtain training data of higher quality. For example, the concept "sailing" was restricted to the category "Travel&Places", such that erroneous content like the music video by Rod Stewart (category "Music") was not downloaded. The YouTube API[9] was used to download videos for each combination of category and concept, obtaining up to 1,000 video clips per concept (this upper limit is imposed by YouTube). For the resulting content, the number of shots per concept was estimated by multiplying the video length with the average number of shots per minute (4.79, estimated on a set of 2,200 YouTube videos). Figure 3.3 plots the estimated number of shots obtained for some random sample concepts. It can be seen that a fair amount of content can be obtained for most concepts (9,031 on average). This quantity is significantly higher than the number of samples used in the TRECVID'08 and Mediamill datasets themselves (1,667 annotations). Some outliers occur: for the concepts "two_people" (21,337 shots) and "meeting" (37,291) many shots can be obtained, and for the concepts "overlayed_text" (401), "waterscape" (569), "Emile_Lahoud" (530), "Duo_Anchor" (107), and "Iyad_Allawi" (988), less than 1,000 shots were found. Generally, this experiment reveals that web video portals offer a sufficient quantity of training content for typical concepts of interest, and that significantly more training content can be acquired than currently used in standard benchmarks. When inspecting the results on a per-concept basis, it also becomes clear that the amount of available material is strongly correlated with user interest. YouTube users tend to upload videos that they find interesting, surprising, funny, or worth presenting otherwise. This can cause problems for certain concepts. For example, the tag "overlayed_text" is given infrequently – though overlayed text appears very often, the tag is simply not used. Other concepts like "waterscape" may not be found interesting enough for filming, editing and uploading videos about them.

3.3.2 Quality of Training Data

While the last section provided a purely quantitative analysis of web video content, it did not address the question whether its *quality* allows a successful concept

[9] http://youtube.com/dev

learning. In the following, an informal discussion on this issue is provided based on a visual inspection of web video content, and key problems for concept learning are pointed out.

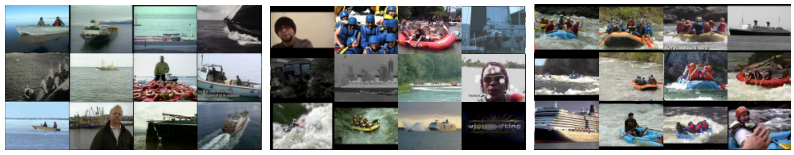
The first and most prominent observation is that web video shows a high variability of production style. Portals like YouTube host almost all kinds of video, ranging from TV snippets to home video. The purpose of this content may be to entertain, to inform, to educate, or even to shock. The target audience can range from a few close friends to a broad world-wide community. Correspondingly, the budget and time invested into the production of a clip may vary significantly, as well as other important parameters such as camera and coding quality. Doubtlessly, all this affects the visual appearance of a video, and with it the concept detectors trained on it.

Second, web tags are coarse: users only provide tags information on clip level, and no shot-accurate label information is given. For example, imagine a user producing a video of his latest sailing trip. Though the video is labeled "sailing", it may also show content that is not visually related to sailing, like trips to towns and nightly parties.

Third, tags are subjective and context-dependent: while standard datasets are annotated according to precise visual criteria, the motivation with which YouTube users assign tags to their clips can be very subtle. For example, a YouTube search for videos tagged with "airplane" returns many shots of airplanes, but also videoblogs about airplane safety, instructions to build paper airplanes, and views from inside an airplane cockpit. Inferring the presence of the concept "airplane" from this visual content may require extra knowledge or may simply be impossible. Obviously, these characteristics of web video content have an influence on concept detectors. Accordingly, web-based concept learning can be characterized as...

- **...A Weakly Supervised Learning Problem**: While the strong annotations used for current concept detection systems guarantee that a concept appears in a shot, web tags are subject to label noise. This is illustrated in Figures 3.4 for the concept "boat_ship": compared to sample frames from a standard dataset (TRECVID'08), training material downloaded from YouTube contains significant amounts of *non-relevant* content. We will address the issue of weak labels more explicitly later (Chapter 4).

- **...A Cross-domain Learning Problem**: Another problem is that web video may differ significantly from the material that concept detection is applied to. For example, concept detectors resulting from training could be applied to news video. In contrast to this target domain, web video



(a) (b) (c)

Figure 3.4: Concept learning from web video as a weakly supervised learning problem and as a cross-domain learning problem: illustrations of training samples for the concept "boat_ship" are given when (a) using a standard training set (TRECVID'07), and (b) using web videos downloaded from YouTube. The web video training set shows significant label noise. (c) Filtered frames from YouTube that have been manually assessed to show the concept. Domain differences between the standard dataset ((a), mostly ships) and YouTube ((c), mostly rafting) can be observed.

is a *mixture* of several video sources, including news video as well as home video, documentaries, etc, and is has been reported previously that significant performance loss is to be expected when generalizing from one domain to another [YH08].

Due to these two reasons, concept learning from web video content can be considered a challenge of significantly higher difficulty than training on manual annotations. Yet, web-based concept detection is appealing due to its benefits regarding scalability and flexibility. Therefore, this chapter will address the question how much performance degredation is to be expected by replacing expensive high-quality datasets with weakly annotated web video content.

3.4 Related Work - Concept Learning from Web Data

Related work on web-based data sources has been targeted at images, video, and also text. For all these modalities, web data provides content at a large scale and interesting application areas — tasks like image and video search, recommender systems, and content filtering could benefit from the automatic inference of semantics. Finally, web data can also be viewed as real-world and unbiased: for example, earlier object recognition benchmarks (which have been criticized as overly simplifying [PBE+06]) have been replaced with data acquired from Flickr [EZWvG06, PBE+06].

Overall, though web-based data has not been exploited to its full potential so far, the research community has recognized the benefits, and visual recognition on web content is an emerging field[10]. First approaches have been developed for visual learning from noisy datasets of web images. For this purpose, topic models have been suggested [FFFPZ05, LWFF07], which identify visual aspects related to objects and separate them from the ones associated with non-relevant images. A similar approach by Yanai and Barnard [YB05] uses Gaussian mixture models over segmented image regions. Pictures acquired from Flickr or image search engines have also been used to complement manually annotated training data for the video domain [CHJ+08], and it has been studied under which conditions this can be successful [KCK06] (for example, if a low number of manually annotated training samples is available). This chapter will demonstrate that a similar use of web video is possible. Beyond this, it will also be shown that we cannot only supplement manually acquired data, but that we should drop a cost-intensive manual annotation for cross-domain concept detection.

Researchers have also investigated the textual annotations of web video and images as a knowledge source. Negoescu and Gatica-Perez [NGP08] applied topic models to image tags to analyze photo groups at Flickr. Based on their model, a keyword search for groups can be realized. Haubold and Natsev [HN08] employed web-based text corpuses for an improved semantic reasoning.

Web video — which is the focus of this chapter — is just beginning to attract researchers' attention. Zelnik-Manor et al. [ZMZP08] and Schindler et al. [SZB08] presented studies on shot boundary detection and categorization of web video content, and emphasized the difficulty of the domain due to enormous content variance and weakness of labels. Chang et al. [CEJ+07] presented a study targeted at consumer video, which also included material downloaded from the web. Yang et al. have presented a multi-modal recommender system for web video [YMH+07]. All this work is targeted at web video as an application domain for video retrieval. This chapter includes similar results for concept detection, but addresses web video not only as an application field but more generally as an information source for concept learning.

[10] IEEE Workshop on Internet Vision, ICME Workshop on Internet Multimedia Search and Mining

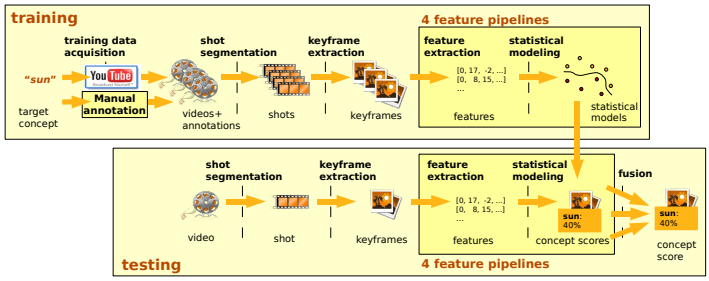


Figure 3.5: The TubeTagger prototype: using training material downloaded from YouTube, models of concept appearance are learned in several *feature pipelines*. These models are then applied to previously unseen videos, obtaining concept-specific scores.

3.5 The *TubeTagger* Prototype

This section describes a prototype that implements the idea of concept learning from web video. The system learns to tag videos by autonomously training on content downloaded from the portal YouTube and has thus been called *TubeTagger*. The key novelty of the approach lies in the data used for concept learning (TubeTagger is the first concept detection system learning from YouTube). Regarding aspects of system architecture, feature representations, and statistical models, best practice in concept detection is followed closely. The system pipeline is illustrated in Figure 3.5. TubeTagger can be run in two modes, one for training concept detection models and one for applying them to previously unseen videos. In training, the system is given a semantic concept by the user and downloads training videos from YouTube (alternatively a conventional manual data acquisition is possible). These videos are preprocessed, i.e. shot boundary detection is performed and keyframes are selected. Keyframes and shots are fed to four *feature pipelines*, each employing visual features of a certain type (for example, color histograms). In each feature pipeline, a supervised classifier is trained, whereas keyframes from videos tagged with the target concept serve as positive samples and frames from all other videos as negative ones.

To detect a target concept in previously unseen videos, the same preprocessing and feature extraction are conducted. Feature-specific scores indicating concept presence are obtained from all keyframes and feature pipelines, and are fused to obtain the final concept score. The system components are described in the order of processing in the following.

3.5.1 Training Data Acquisition

TubeTagger can be run in two modes of training data acquisition: first, the user can provide video data with manually acquired annotations, similar to other concept detection systems. This setup will be used for quantitative comparisons in later experiments.

Alternatively, TubeTagger can contact YouTube for training material and derive class labels from user-generated tags. In this case, only a textual description of the target concept must be provided. Optionally, the quality of training material can be improved using the fact that videos at YouTube are organized in categories such as "Sports", "Travel&Places", or "People&Blogs". This is done by restricting video downloads to a certain category.

3.5.2 Shot Segmentation and Keyframe Extraction

Each input video is segmented into shots, and for each shot representative keyframes are extracted. This reduces data load significantly but also causes a certain information loss. Correspondingly, keyframe selection should adapt to the content of a video: for long shots containing strong scene activity, multiple keyframes might be appropriate, while short or static scenes can be represented by a single frame.

For this purpose, an adaptive two-step procedure is used. First, the video is segmented into shots by thresholding differences of MPEG-7 color layout descriptors (CLDs) [MOVY01]. Second, for each of the resulting shots, a clustering is applied similar to the one by Hammoud and Mohr [HM00]: a Gaussian mixture model is fitted using a K-Means clustering over frames. For each mixture component, the frame next to the cluster center is extracted as a keyframe. The optimal number of components is determined using the Bayesian Information Criterion (BIC) [Sch78]. The resulting method gives $1-5$ frames per shot depending on the visual content (an average of 1.75 was estimated on a set of $2,200$ video clips).

3.5.3 Feature Pipelines

Like most concept detection systems [WLL+07, YCKH07, Sno07], TubeTagger employs several visual features and statistical models. Four types of features and models are integrated in *feature pipelines* $F_1, ..., F_4$. Each feature pipeline F_j represents a type of visual feature (like color histograms) and gives a specific score $P_{F_j}(t|x_i)$ for each keyframe x_i. The feature pipelines are outlined in the following:

Pipeline 1 - Visual Words (SIFT)+SVM This patch-based approach uses the popular *bag-of-visual-words* representation [DKN05, FFFPZ05, SZ03], which clusters local features according to their appearance into patch categories called *visual words*. From an input image I, a set of local patches is sampled using an interest point detector [SMB00] or random or regular sampling [NJT06]. Patch are represented by local descriptors $f_1, ..., f_K$, which are matched with a codebook of representative patch prototypes $f'_1, ..., f'_m$ obtained from a clustering of patch descriptors. This gives a sequence of patch category entries (so-called *visual words*) $c_1, ..., c_K$:

$$c_k = \underset{j=1,...,m}{\arg\min} ||f_k - f'_j||_2$$

The frequency with which visual words appear in I is stored in a histogram $x^1, ..., x^m$, the so-called *bag-of-visual-words* feature:

$$x^j = \sum_{i=1}^{n} \delta(j, c_i)$$

The model draws an analogy to the well-known *bag-of-words* model from text retrieval [Lew98]. It provides a good tradeoff between a robust description on the one hand and computational feasibility on the other. Some sample visual words are illustrated in Figure 3.6(a): it can be seen that the patches belonging to a visual word share a common appearance, and sometimes also coherent semantics: for example, some visual words tend to contain parts of faces (line 4) or facets of the horizon (line 5). Obviously, their presence in an image is an indicator of semantic concepts such as "interview" or "outdoor".

By combining bag-of-visual-words features with SVMs [SS01] as a statistical model, recognition systems have been very successful in a variety of visual recognition tasks, like object category recognition [EVGW+07], scene categorization [LSP06, QMO+07], or the filtering of pornography [DPN08]. The method has also given excellent results on a standard concept detection benchmark [vdSGS08].

This bag-of-visual-words approach is also adopted in the TubeTagger framework. Since it has been demonstrated that performance is strongly correlated with the number of patches per image [NJT06], a dense regular sampling at several scales is done that gives a large number of $3,600$ patches per frame. Each patch is described by its 128-dimensional SIFT representation [Low04], which consists of localized gradient direction histograms over the patch area. Optionally, SIFT descriptors can also achieve rotation invariance by normalizing patches to a canonical angle. However, this normalization is omitted here as many concepts tend to

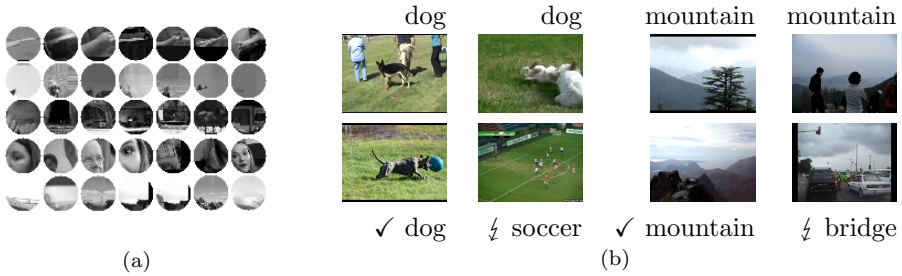


Figure 3.6: (a) Visual words from a codebook of visual words (sample patches in a line belong to the same cluster). Some clusters can be associated with certain semantics (for example, they tend to contain parts of faces or of the horizon). (b) Sample results of nearest neighbor matching: two matches for the concepts "dog" and "mountain". Query frames are in the top row, nearest neighbors in the bottom one (pictures from YouTube).

come with a characteristic angle and additional invariance reduces the discriminative power of features. The resulting patches are mapped to a $2,000$-dimensional codebook previously trained using a K-Means clustering.

For each target concept, a two-class SVM is trained using the libsvm implementation [CL01]. As a kernel function, the χ^2 kernel is used, which has empirically been demonstrated to be a good choice for visual word features [ZMLS07]:

$$K(x,y) = e^{-\frac{d_{\chi^2}(x,y)^2}{\gamma^2}}.$$

The scale parameter γ is estimated using cross-validation. d_{χ^2} is the χ^2 distance between visual word histograms x and y:

$$d_{\chi^2}(x,y) = \sum_{i=1}^{m} \frac{(x^i - y^i)^2}{x^i + y^i}$$

Pipeline 2 - Light-weight Visual Words (DCT) + SVM Like the first pipeline, this one uses a bag-of-visual-words approach. However, since the extraction of a high number of SIFT features is time-consuming (the standard implementation[11] requires about 15 seconds per frame), a light-weight alternative is investigated based on DCT coefficients associated with video macroblocks. This information is fast to compute (and is alternatively available in the compressed video

[11] http://www.robots.ox.ac.uk/ vgg/research/affine/

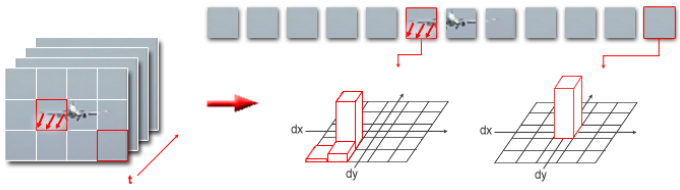


Figure 3.7: An illustration of motion features: frames are divided into tiles. For each tile, a histogram of MPEG-4 motion vectors is stored, and all histograms are concatenated to a motion descriptor. Two histograms are illustrated: one captures the bottom-left motion of the airplane, one the absence of motion in a background tile.

stream, such that only a partial decompression is required). Patches of size 16×16 pixels are sampled at regular steps of size 16, which gives 234 patches per frame. These are described by low-frequency Discrete Cosine Transform (DCT) coefficients in YUV color space. A motivation for this description is given in [Kö03], where DCT base functions are demonstrated to show an strong resemblance with principal components learned from natural images. The DCT representation can thus be interpreted as an approximation to Principal Component Analysis (PCA) [DHS00] without requiring an additional training step. 78 coefficients are extracted for each block in a zigzag pattern, 36 for the intensity and 21 for each chroma component. Like in Pipeline 1, a vocabulary of $2,000$ visual words is learned using K-Means, and SVMs with a χ^2 kernel are used as a statistical model.

Pipeline 3 - Color and Texture ("CT") This feature pipeline uses global frame-level descriptors based on color and texture. Color is represented by RGB histograms with 8^3 bins, and texture by similar histograms over the Tamura texture properties *coarseness*, *contrast*, and *directionality* [TMY78]. Both features are combined using early fusion (i.e. concatenated), obtaining a joint 1024-dimensional feature vector. As a statistical model, nearest neighbor matching is used as illustrated in Figure 3.6: given a keyframe x_i and a training set of labeled keyframes Y, we find the nearest neighbor $x'_i := \arg min_{y \in Y} ||y - x_i||_2$, and the score for a concept t equals a vote for the tag of this neighbor. To realize fast matching, an approximate search with a kd-tree is used [PPC01]:

$$P_{F_3}(t|x_i) := \delta(t, t(x'_i))$$

Pipeline 4 - MPEG Motion Vector Histograms ("Motion"): For some concepts, motion can be a more appropriate representation than color or texture. For example, while the appearance of frames showing the concept "interview" may vary strongly, interviews may be characterized well by the fact that the interviewee in the frame center makes occasional gestures, while the background remains static. A simple feature of MPEG-4 block motion vectors extracted by the codec XViD[12] is used to describe *what* motion occurs as well as *where* it occurs. The spatial domain is divided into 4×3 regular tiles, and for each tile a two-dimensional 7×7 histogram is computed over the 2D components of all motion vectors in the tile (vectors are clipped to $[-20, 20] \times [-20, 20]$). By concatenating those histograms, a 588-dimensional descriptor is extracted on shot level. For an illustration, see Figure 3.7. Like for color and texture, nearest neighbor matching is used as a statistical model.

3.5.4 Fusion

From several keyframes and feature pipelines, weak pieces of evidence are obtained indicating the presence of semantic concepts. These are fused in two steps to obtain the final concept score $P(t|X)$. First, a fusion over the keyframes $x_1, ..., x_n$ of a video X is done using the well-known sum rule from classifier combination:

$$P_{F_j}(t|X) = \frac{1}{n} \sum_{i=1}^{n} P_{F_j}(t|x_i)$$

This approach outperformed other standard fusion methods (like the max, product, and min rule) in previous tests, which confirms earlier theoretical results that claim a good robustness with respect to noise in the input scores [KHDM98]. Such robustness is crucial in the context of web video tagging, since many keyframes may not be visually related to the target concept and thus give misleading scores.

Second, to combine scores obtained from several feature pipelines, a range normalization of all scores to $[0, 1]$ is applied [NNT05], and the normalized scores are combined using a *weighted sum* fusion:

$$P(t|X) = \sum_{j=1}^{4} w_j P_{F_j}(t|X). \tag{3.1}$$

The feature weights $(w_1, w_2, w_3, w_4) \in [0, 1]^4$ are learned by a grid search optimization on a validation set (the same weights were used for all concepts).

[12] www.xvid.org

Table 3.1: The 22 concepts of the *Youtube-22Concepts* dataset. Each concept is assigned a canonical YouTube category to refine the downloaded material.

concept	youtube category	concept	youtube category
basketball	Sports	hiking	Travel&Places
beach	Travel&Places	interview	News&Politics
cats	Pets&Animals	race	Autos&Vehicles
concert	Music	riot	News&Politics
crash	Autos&Vehicles	sailing	Travel&Places
dancing	People&Blogs	secondlife	Gadgets&Games
desert	Travel&Places	soccer	Sports
eiffeltower	Travel&Places	swimming	Sports
explosion	How-to&Do-it-yourself	talkshow	People&Blogs
golf	Sports	tank	Autos&Vehicles
helicopter	Autos&Vehicles	videoblog	People&Blogs

3.6 Experiments

In this section, experiments with the TubeTagger prototype are presented in which the challenge of concept learning from web video is addressed. Two experiments are conducted in which TubeTagger is trained on YouTube material:

1. **Testing on Web Videos**: First, TubeTagger is both trained on and applied to web video content. When used in this scenario, concept detection can support an automatic content-based indexing and search for web video.

2. **Testing on Other Domains**: This experiment addresses the question whether concept detection systems trained on web video can be applied successfully to different domains. Here, the system trained on YouTube is tested on news video and documentary TV, and comparisons with systems trained on manually annotated standard datasets are provided.

3.6.1 Experiment 1 - Web Video

In a first experiment, the TubeTagger prototype is both trained on and applied to web video content. The purpose of this experiment is to give a first impression of the performance that can be achieved when learning concepts from web videos, and to provide a quantitative comparison of several types of visual features. Tests are performed on a dataset of real-world web videos downloaded from YouTube, which is described first. After this, the experimental setup is outlined, and quantitative results are presented.

Figure 3.8: Youtube users produce series of videos sharing a common production style. These keyframes are sampled from different clips but show the same actors and similar overlayed text. The fact that concept detection systems overfit to such redundant material leads to biased benchmarking results (pictures from YouTube).

The Youtube-22Concepts Dataset A web video dataset was collected by downloading YouTube clips for 22 semantic concepts. The concepts were manually chosen, and no standard concept list like LSCOM or Mediamill was used as a basis (comparisons with such standard concepts will be outlined in Section 3.6.2). The following standard criteria [NST+06] were taken into account when selecting concepts: (1) *feasibility* — the semantic concepts should be inferable from the visual video content. Abstract terms like "love" or "humor" were excluded. (2) *coverage* — a variety of concepts should be included, like locations (e.g., "desert", "beach"), activities (e.g., "hiking", "interview"), objects (e.g., "cat", "eiffeltower"), and sports (e.g., "swimming", "soccer"). (3) *availability* — visually related content should be available, which was briefly checked by inspecting the first pages of a YouTube search result. Each tag was assigned to a canonical YouTube category to improve the quality of downloaded material. See Table 3.1 for a complete list of all 22 concepts and categories.

The dataset was downloaded in summer 2007. The top 100 videos were downloaded for each tag, obtaining a database of 2,200 clips with a total length of about 194 hours. The whole set was separated into a training set (50 videos per concept), validation set, and test set (both 25 videos per concept). This dataset is referred to as the **Youtube-22Concepts** dataset in the following. It has been made available on request[13] to support research on web video retrieval (including video data, YouTube URLs, and all meta-data available).

The Role of Redundancy Videos at YouTube contain lots of redundant content uploaded multiple times. While some forms of redundancy are easy to identify

[13] http://tagmyvids.com/project.html

(like duplicate videos), others are subtle and difficult to uncover automatically. Four levels of redundancy can be identified, listed in the order of an increasingly difficult automatic detection:

- **Exact Duplicates**: Videos that have been uploaded multiple times, whereas slight differences of coding quality may occur.
- **Near-duplicates**: Users upload the same video but modify it slightly (for example by adding or replacing title strips).
- **Shot Re-use**: Users recompile popular video scenes, which appear multiple times in collections like "Funniest Soccer Moves" or "Best Harry Potter Moments". Here, redundant content appears in different combinations and compositions.
- **Series**: Like TV, YouTube hosts actual series of video clips sharing a homogeneous production style (for an example, see Figure 3.8).

Obviously, such redundancy has an influence on benchmarking results: if the same content appears in both training and testing, a concept detection system having learned the training version can easily assign the correct tag to the test version. Note that in some scenarios, this influence is *wanted*: for example, if tagging a users' individual video collection, it seems reasonable to exploit redundancy for a personalized tagging. When targeted at measuring the performance of general-purpose concept detection systems, however, evaluation results may be biased by the influence of redundancy, and duplicates should be removed as far as possible. While this issue is not taken into account in the TRECVID campaign, at least the first two kinds of redundant material were eliminated here. For this purpose, a two-step procedure was used. First, duplicates were identified automatically by matching clip *signatures* of color and motion with an edit distance [ZT06]. Second, near-duplicates were identified as they typically caused suspiciously good concept scores, i.e. top-ranked clips were compared with their nearest neighbors in the dataset and were eliminated manually if appropriate (on average, ca. 5 videos per concept were removed this way).

Performance Measure As a performance measure, average precision (AP) is used, which is standard practice in concept detection research [KO08]. Videos are sorted by descending concept score, obtaining a *ranked retrieval list* $x_1, ..., x_n$ with ground truth labels $y_1, ..., y_n \in \{-1, 1\}$. If this list is thresholded at rank T, the retrieval system only returns videos $x_1, ..., x_T$. In this set of retrieved items, we

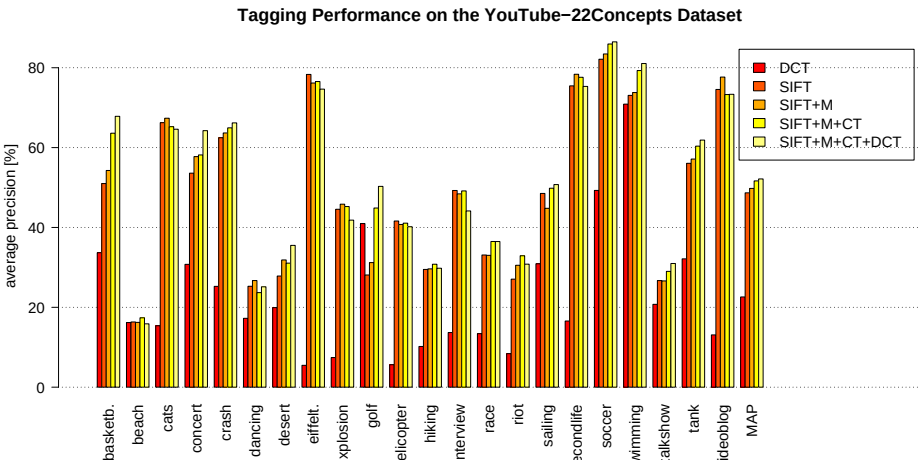


Figure 3.9: Quantitative results of YouTube tagging. A patch-based approach using SIFT features achieves a high performance, which can be improved further using additional pipelines such as color+texture and motion.

assume to have r_T relevant items, and N relevant items in the whole collection. Then *recall* R_T and *precision* P_T are defined as quality measures of a retrieval result:

$$\begin{aligned} P_T &= r_T/T \\ R_T &= r_T/N \end{aligned}$$

Recall measures the coverage of the system's feedback, whereas precision measures its purity. Ideally, $P_T = R_T = 1$. By thresholding at all positions of the ranked retrieval list corresponding to relevant items, the *recall-precision curve* is obtained. The average precision (AP) corresponds to the area under this curve:

$$AP = \frac{1}{N} \sum_{T:y_T=1} P_T \tag{3.2}$$

The average precision measures the quality of the retrieval result for a single concept. To obtain an overall performance measure, the *mean* average precision (MAP) over all concepts is used.

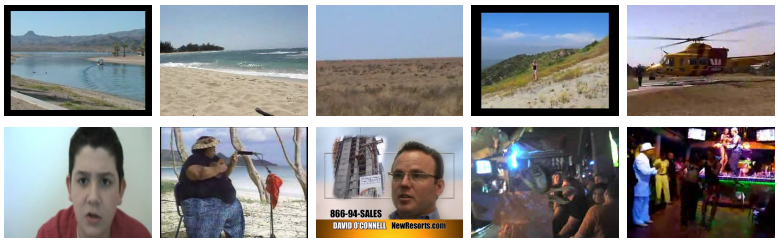
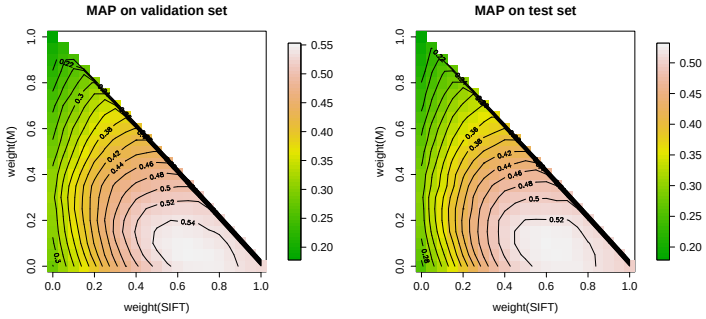


Figure 3.10: Detecting the difficult concept "beach". Top: YouTube clips that give the highest scores. Bottom: false negatives, i.e. beach clips with the lowest scores. Only one clip is visually related to the target concept (pictures from YouTube).

System Parameters The TubeTagger system was run for all four pipelines described in Section 3.5: SIFT visual words with SVMs (*SIFT*), their light-weight DCT equivalent (*DCT*), as well as motion (*M*) and color+texture (*CT*) with nearest neighbor matching. SVM training requires the estimation of the parameter γ (Equation (3.5.3)), and of a cost parameter C associated with training sample misclassification (please refer to [SS01] for more information). Both parameters were estimated using a grid search maximizing the cross-validated average precision [HCL03]. A practical problem is that the training sets in concept detection are often extremely *imbalanced*, i.e. the number of negative samples usually outnumbers the number of positive ones by far. This causes difficulties for many classifiers, including SVMs [AKJ04]. To overcome this problem, the dominant class was subsampled to obtain a roughly balanced training set, with the number of negative samples fixed to $6,000$.

Results Quantitative results in Figure 3.9 reveal that concept detection in web video is challenging but feasible in general. A mean average precision of 52.2% ($n = 12,100$) is achieved for a system that combines all four feature pipelines, which is a 11.6-fold improvement over a random sorting of the ranked retrieval list (4.5%). An inspection of the single feature pipelines reveals a dominance of the SIFT+SVM approach, which outperforms light-weight DCT visual words significantly and gives a mean average precision of 48.7% (DCT by itself gives 26.1%). Adding other features leads to further performance improvements (1.1% for motion, 1.9% for color and texture, 0.5% for DCT visual words), which are moderate but significant according to a sign test over the rank improvement of positive items (level 99%). Overall, this confirms earlier results, which report an excellent perfor-



(a) (b)

Figure 3.11: The concept detection performance (MAP) of TubeTagger plotted against the weights for SIFT+SVM and Motion. Similar performance can be observed on the validation set (a) and on the test set (b), which indicates that feature weights can be learned reliably.

mance for visual word approaches for the domain of news video [vdSGS08] — obviously, similar observations hold for web video content. Exceptions are some concepts for which the color+texture pipeline improves performance strongly. These are mostly sports concepts, like "golf", "basketball", and "swimming", for which color is obviously a strong clue. Also, sports-related concepts reach the highest overall performance, obviously because they come with a static global frame layout and low intra-concept variance (an average precision of up to 86.5% ["soccer"] is reached). In contrast to this, for the most difficult concept "beach" only 15.9% are achieved. A closer look at this concept is taken in Figure 3.10, including the 5 "beach" videos with the lowest scores (*false negatives*). Only one of these false negatives is visually related to the tag "beach", while the others show videoblogs tagged with "beach" and nightly parties in Miami Beach. Obviously, the reason for system failure in these cases is that the relation between a clip and its tags is subtle and extraordinarily difficult to infer from the visual content only.

To test whether the feature weights $w_1, ..., w_4$ (Equation (3.1)) can be learned reliably on the validation set, performance is plotted in Figure 3.11, both for the validation set and the test set (note that DCT features were left out, and that the weight of the C+T pipeline adds up to 1). Though the performance is lower on the test set, a similar behavior of tagging performance can be observed for validation and testing, which indicates that feature weights can be learned reliably.

Table 3.2: The concepts used in Experiment 2. These are 19 of the 20 concepts used in the TRECVID'08 benchmark. Columns 2,3,5, and 6 provide details on how information was downloaded from YouTube.

concept	YouTube query	YouTube category	concept	YouTube query	YouTube category
Classroom	classroom & school -secret	-	Telephone	phone & device	-
Bridge	bridge -crossing -ship	Travel&Places	Street	street & paved	-
Em._Vehicle	emergency & vehicle -driver -ride	Autos&Vehicles	Demonstr.	protesting	-
Dog	dog	Pets&Animals	Hand	hand & daft	-
Kitchen	kitchen -knife -remodel	Howto&Style	Mountain	mountain &panorama	Travel&Places
Airplane_fl.	airplane & flying -jefferson -indoor -school -kids	Autos&Vehicles	Nighttime	by & night	Travel&Places
Bus	bus -van -suv -vw -ride	Autos&Vehicles	Boat_Ship	ship & (queen | freedom | royal)	Autos&Vehicles
Driver	car & vehicle & driver -simulator	Autos&Vehicles	Flower	flower & (bouquet | bloom)	-
Cityscape	cityscape -slideshow -emakina	Travel&Places	Singing	singing & (gospel | choire)	-
Harbor	harbor & industry & ship	-			

3.6.2 Experiment 2 - Other Domains

In Experiment 1, the TubeTagger prototype has been both trained on and applied to web video content downloaded from the portal YouTube, and the general feasibility of concept learning from web video has been demonstrated. Yet, a fundamental key question remains unanswered, namely how taggers trained on web video perform on other domains.

To answer this questions, the following experiment provides performance comparisons for the TubeTagger prototype when training and testing on web video material as well as standard datasets of news and documentary TV from the TRECVID benchmark. This experiment requires video data and corresponding concept annotations, which are available for several video sources and concepts from joint efforts of the research community. In the following, these datasets are described:

Figure 3.12: The top 5 detections of the YouTube-based system for several concepts on the TRECVID'07 dataset. While the system works well for some concepts ("mountain", "cityscape"), it suffers from a mismatch between YouTube and the target domain for others ("telephone").

Concepts Since for some of the concepts from the *Youtube-22Concepts* dataset no publicly available annotations exist, a different set was used, namely 19 of the 20 concepts used in the TRECVID'08 benchmark (the concept "two_people" was omitted, as one of the video datasets used lacked publicly available annotations). These concepts stem from the LSCOM ontology for multimedia retrieval and can thus be considered a standard test case. The full list of concepts can be found in Table 3.2. For detailed descriptions, please refer to the LSCOM website [LSC].

Video Data and Annotations Three video datasets were used in this experiment, whereas corresponding annotations were acquired from publicly available standard datasets or from YouTube:

1. **TV05**: This dataset, used in the TRECVID'05 "High-level Feature Extraction" task, is the most frequently used test dataset for concept detection, and extensive manual annotations are available [KHN+06, SWvG+06]. The dataset contains video data from 13 news programmes, including US, Chinese, and Arabic broadcast [OIKS05]. It consists of a predefined development set of 86 hours and test set of 85 hours. Annotations for the 19 test concepts were downloaded from the LSCOM website [LSC]. As these cover only the

development set, the test set was neglected, and the development set was split into a training set and test set of equal size (the split was done between different broadcast dates). Keyframes were extracted using the adaptive approach described in Section 3.5. To reduce data load, only one keyframe per shot was kept, which gave a total of 75,000.

2. **TV07**: In 2007, TRECVID's "High-level Feature Extraction" task used data provided by the Netherlands Institute of Sound and Vision [KO07]. This dataset contains news magazines, science news, news reports, documentaries, educational programming, and archival video, in a development set and test set of 50 hours each. Annotations have been acquired at the Chinese Academy of Sciences and were used by participants in TRECVID'08. The adaptive keyframe extraction approach described in Section 3.5 was used, which resulted in about 113,000 keyframes.

3. **YOUTUBE**: Like in the previous experiment, a dataset of videos downloaded from YouTube was used. 100 clips were acquired for each concept. Only short videos of up to 3 minutes length were used to reduce data load. Queries were also manually refined to guarantee a certain quality of the resulting content (for example, the query "classroom" was replaced with "classroom+school" to obtain content that is closer to the description of the original TRECVID feature). This refinement was done without knowledge of concept appearance in the TRECVID video data. Like in the first YouTube experiment, YouTube search results were also filtered by category. A full list of the YouTube queries used can be found in Table 3.2. The dataset has a total length of about 42 hours, which corresponds to 36,000 keyframes extracted using the adaptive approach from Section 3.5. To get a broader coverage of negative samples, 30,000 frames from the *Youtube-22Concepts* dataset were added, obtaining an overall of 66,000 frames.

Setup The results of Experiment 1 indicate that a visual words approach provides a high performance, and that by fusing it with other features moderate performance improvement can be achieved. For the sake of computational efficiency — and since the focus of this experiment is on the *relative* performance when training on different data sources — in this experiment only the best feature pipeline (SIFT visual words) was used. Also, SIFT descriptors were replaced with 128-dimensional SURF features [BTvG06], a faster approximation. Again, a fixed number of samples was used for the negative class to roughly balance the training set — for the TRECVID'07 and TRECVID'05 datasets with fewer annotations

Table 3.3: Concept detection performance when training and testing on YouTube and two TRECVID datasets (TV05 and TV07). The detectors trained on the target domain outperform the others significantly ($n = 573,154$ [TV05], $344,508$ [TV07], $11,913$ [YOUTUBE]).

MAP[%]			
training / testing	TV05	TV07	YOUTUBE
TV05	**18.40**	3.82	14.68
TV07	3.32	**9.65**	16.49
YOUTUBE	2.83	3.51	**31.33**

(median: 291/273 annotations, minimum: 22 for "airplane" / 51 for "kitchen"), the number of negative samples was set to $1,500$. For the youtube data (median: 1332, minimum 374 for "mountain"), it was set to $6,000$.

Results The TubeTagger engine was trained on all three training sets, obtaining three different statistical models for each of the 19 target concepts: two standard models trained on the manually annotated TV05 and TV07 training sets, and one trained on videos downloaded from YouTube. Each of the three models was applied to all three test sets, obtaining 9 concept detection runs. These were evaluated to investigate (1) which model performs best for concept detection in web video, (2) which training sets lead to the best performance on the TV05 and TV07 standard test sets, and (3) how the YouTube-based tagger compares to state-of-the-art detectors when generalizing to a domain unseen in training.

Some sample results of YouTube-based detectors on the TV07 test set are illustrated in Figure 3.12. It can be seen that the system works well for some concepts (like "mountain"), while for others (like "telephone") no hits are found. An in-depth inspection revealed that this difference is caused by a strong variation of the quality of YouTube training material: while for "mountain" lots of panoramic mountain views were obtained, the "telephone" training set tends to show close-ups of the latest smartphone gadgets, and correspondingly computer screens and similar structures are detected. This leads to a poor result, as "telephone" scenes in the TV07 test set show mostly phones on office desks.

Quantitative results are provided in Figure 3.14, and the mean average precision for all runs is also given in Table 3.3. Let us first study concept detection on the YouTube test set. Here, it can be seen that the YouTube-based system outperforms the two standard detectors (MAP 31.33% compared to 14.68% and 16.49%). This indicates that for tagging YouTube videos, YouTube as a training set, as could be expected, outperforms standard datasets.

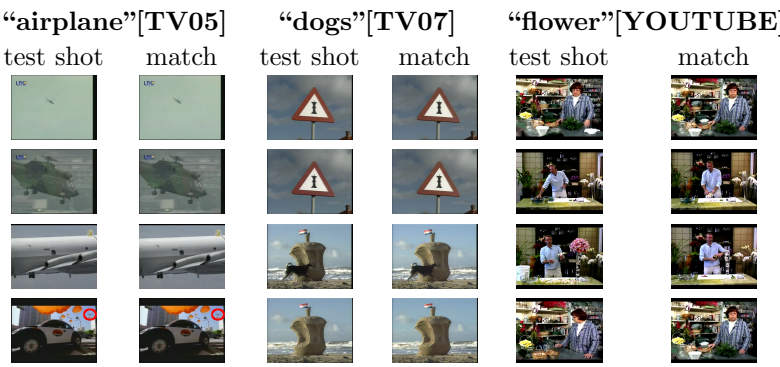


Figure 3.13: Specialized detectors trained on the target domain significantly outperform all others. One reason for this are duplicates in the datasets: these keyframes are from the shots giving the highest scores for the top-rated concepts in TV05 (“airplane”), TV07 (“dogs”), and YOUTUBE (“flower”), together with nearest neighbors from the training set. For all these test frames, a (near-) duplicate is found in the training set (note that “airplane” shot no. 4 shows the object only at a very small scale).

More generally, it can be seen that each detector performs best on the data source it was trained on, i.e. on TV05 and TV07 the YouTube-based tagger is significantly outperformed by the “specialized” detectors trained on the corresponding training sets. This indicates that the acquisition of manual annotations on the target domain improves performance significantly. An in-depth analysis reveals that one reason for this dominance of the specialized detectors is redundancy. This is illustrated in Figure 3.13 for the concepts achieving the best performance on each of the test sets. In all cases, the 4 shots achieving highest scores are illustrated on the left, and their nearest neighbors from the training set on the right. Obviously, all concept detectors implicitly match test content with (near-) duplicates in the training set: for “airplane” in TV05, matches include very small airplanes in the background that could probably not be detected without redundancy. For “dogs” in TV08, the system overfits to a single shot including a street sign and a dog. For “flower” in YOUTUBE, the system makes use of two series about flower arrangements. It is obvious that redundancy leads to biased and overly positive benchmarking results. Yet, it should also be noted that — if duplicates are not filtered — this effect cannot be separated from other factors like production style.

Finally, the YouTube-based detector is compared with standard ones (TV05 / TV07) when testing both on a third, novel data source (TV07 / TV05). We can

observe that all systems generalize poorly to domains not seen in training, which confirms earlier results by Yang and Hauptmann indicating that concept detectors strongly overfit to the domain they are trained on [YH08]. Also, results reveal that the YouTube-based detector generalizes only slightly worse to novel domains than the standard detectors. On the TV07 test set, a performance of 3.51% is achieved compared to 3.82% (training on TV05). On the TV05 test set, 2.83% are achieved compared to 3.32% (training on TV07). This corresponds to a moderate relative performance loss of 11.4% compared to a cost-intensive manual annotations.

The conclusion I draw from this experiment is that if annotations on the target domain are available, they definitely help to increase performance. However, if this is not the case, concept detectors perform poorly for *any* training sets, i.e. also if using manual annotations. Here, training on YouTube (which can be done without any manual annotation effort) offers an appealing alternative.

In a final test, the question is addressed whether *enriching* standard training sets with material downloaded from YouTube leads to an improved generalization of concept detection. Therefore, the TV05 and TV07 training data were combined with YOUTUBE samples. The concept detectors trained on these joint training sets were then applied to third domain unseen in training (TV05 in case of TV07+YOUTUBE, TV07 in case of TV05+YOUTUBE). The average precision achieved in both cases is plotted in Figure 3.15: by adding YouTube data to the training set, in one case (when testing on TV05), only a minor improvement from 3.32 to 3.36% is achieved. In the other case (when testing on TV07), mean average precision is improved from 3.82% to 4.67%. On average, this corresponds to a relative performance improvement of 11.7%, which is significant according to a sign test over the rank improvement (level 99%). This shows that additional training data from YouTube can help concept detection systems to generalize better to novel domains.

3.7 Discussion

The effort associated with the manual acquisition of training annotations is a key problem with respect to the practical application of concept detection, as it limits the size of detector vocabularies and the adaptation to changes in users' information needs. To overcome this problem, it has been proposed in this chapter to learn concepts from user-tagged web video. This setup allows a scalable and flexible concept learning, as class labels can be derived automatically from user-generated tags.

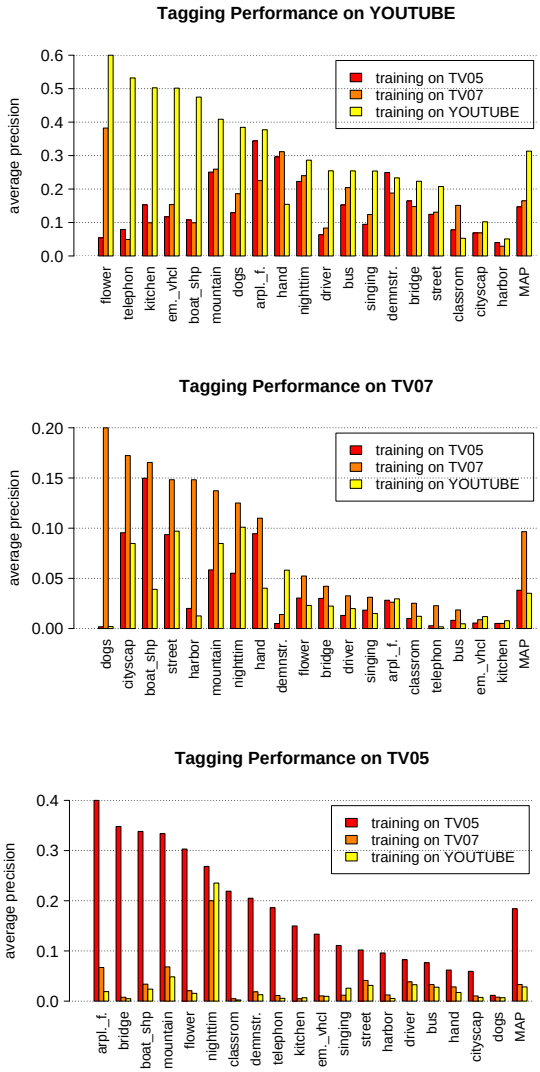


Figure 3.14: Quantitative results of concept detection when (a) testing on YOUTUBE, (b) testing on TV07, and (c) testing on TV05. All detectors perform best on the domain they are trained on, and generalize poorly to different datasets. On domains unseen in training, the YouTube-based tagger performs comparable to a training on manually annotated standard data.

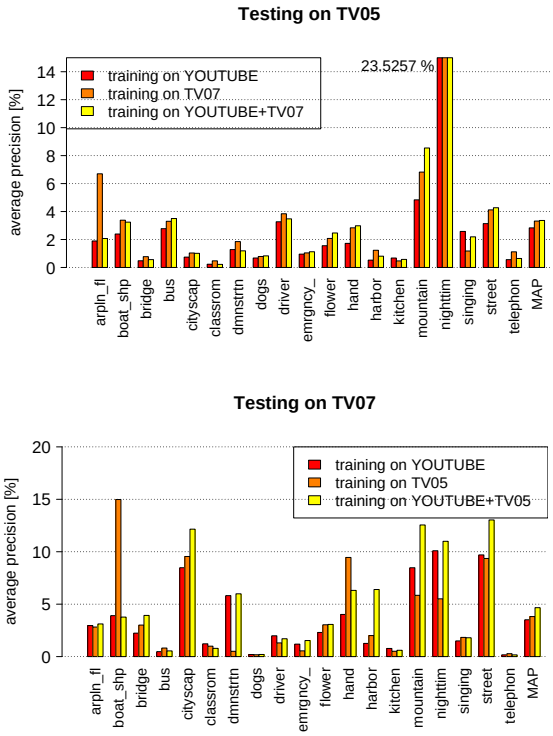


Figure 3.15: Quantitative results of cross-domain concept detection. By enriching standard training sets with material from YouTube, the generalization performance of concept detectors can be increased.

A concept detection prototype named *TubeTagger* has been presented, which is capable of using YouTube data directly as training material. Our experimental results with the system have first shown that the resulting detectors work well when applied on the same domain as they were trained on (i.e., YouTube). This shows that visual learning from web video is possible in general. We have then applied TubeTagger to datasets from the TRECVID campaign. It could be seen that, on these different domains, YouTube-based detectors are outperformed by "specialized" systems trained directly on the target domain. Though this was to some extent caused by the presence of duplicate material in the benchmark datasets, the result suggests that – if manually acquired ground truth on the target domain is available – it is to be preferred over YouTube-based training.

However, the situation was found to be different when generalizing to domains unseen in training. Here, significant overfitting was a key problem for all systems, and surprisingly detectors trained on web video performed about as well as the ones trained on strongly annotated data. Also, detection rates could be improved in this situation by supplementing conventional training sets with YouTube content.

Overall, these results demonstrate that web video is a highly interesting data source for concept detector training. With large-scale readily annotated data offered by services like YouTube, concept detection systems can be trained under less supervision, can scale up to more concepts, and thus provide better support for video search. Compared to the proposed web-based concept learning, a manual annotation of training sets may not really be worth the effort, as it only gives improvements on the restricted training domain. For a practical application in which a concept detector is applied to video sources unseen in training, it seems preferable to automatically bootstrap detection from web video and then perform a light-weight manual refinement on the target domain, for example using relevance feedback [RL03].

Chapter 4

Relevance Filtering for Weakly Labeled Video

In this chapter, the idea of concept learning from web video is pursued further. The problem of label noise is addressed, which refers to the fact that user-generated tags are subjective and coarse, such that training examples contain non-relevant material. To overcome this problem, a novel approach called *relevance filtering* is suggested, which views web tags as weak indicators of true, latent class labels. During concept learning, these labels are inferred, and non-relevant training content is discarded. The key contributions of this chapter are[1]:

1. It is demonstrated that web video training sets show significant label noise — typically, only $20 - 50\%$ of training content downloaded from YouTube do actually show the target concept ($n = 89,500$).

2. Also, it is shown that this label noise degrades the performance of detectors severely, with a relative performance loss of up to 33% ($n = 100,000$).

3. A novel approach called *relevance filtering* is suggested, which combines concept learning with an elimination of non-relevant training content. This approach can be integrated with a variety of supervised learning techniques, as is demonstrated for kernel densities and Support Vector Machines.

4. Relevance filtering is shown to improve the robustness of concept learning with respect to label noise. In several experiments on YouTube content,

[1] This chapter is based on the author's work in [BUB09, UBB09, USB08, USKB08b]

relative performance improvements of up to 17% are reached compared to standard supervised learning ($n = 100,000$).

4.1 Introduction

Recent technological developments like high-speed internet and increasing storage capacity have made it possible for private users to generate, publish, and share large amounts of digital video [Sme07, USA06]. To realize efficient data access, the most frequently used strategy is a text-based search on the basis of keywords. This, however, requires an *indexing* that links the video content in a database with semantic concepts (or *tags*) appearing in it, like objects ("airplane"), scene types ("cityscape"), and activities taking place ("interview"). The current approach of constructing such an index — as followed by most video-sharing portals — is based on filenames, user-generated tags, and other meta-information. The problem with this is that meta-data is often not at hand, and manual annotations are subjective, incomplete, coarse, or too time-consuming to acquire.

This opens the question whether an automatic indexing at a large scale can be bootstrapped from a limited amount of manual annotations. This strategy is followed by *concept detection* systems, which use content-based features and supervised machine learning techniques to construct statistical models for the appearance of concepts. Afterwards, these models are used to infer the presence of concepts automatically in previously unseen content [CHL+07, Sno07, WLL+07, YCKH07]. Though such detectors do not reach an accuracy comparable to a careful manual annotation, they have been demonstrated to be very useful in a video search context [SWdR+08].

Yet, an application of concept detection in practical large-scale settings has not been realized. A key problem lies in the manual annotation effort associated with training data acquisition: the number of potential target concepts is high (usually in the range of thousands), and new concepts of interest emerge constantly (like "financial crisis" or "olympics 2008"). In contrast to this, current prototypes utilize only a few hundred static concepts [NST+06, YCKH07] simply because the manual acquisition of training information is time-consuming and cost-intensive.

In Chapter 3, we have already studied an interesting option to overcome this scalability problem: web video was investigated as a novel information source for concept learning, and explicit manual annotations — which are precise but difficult to acquire — were substituted with label information automatically derived from user-generated tags. It was demonstrated that a more scalable and flexible

concept learning can be realized this way, and that the resulting detectors generalize comparably well to new domains as the ones trained on manually acquired standard data. This suggests a novel way of bootstrapping concept detection from web video.

In this chapter, I will elaborate on this idea further. The goal is to learn concepts in which certain objects or scene types are visually present, like "Eiffel Tower" ("scenes showing the Eiffel Tower") or "desert" ("scenes showing desert landscape"). To achieve this, content with corresponding tags is downloaded from YouTube and serves as training material.

To improve such concept learning, I will address the fact that web-based content comes with significant *label noise*, i.e. material annotated with a target concept *may* show it but does not necessarily do so. This is due to several reasons: first of all, tags are coarse and indicate *that* a concept appears in a video clip, but not *when* it appears. Imagine a YouTube clip of Paris that shows a shot of the "Eiffel Tower" (and is correspondingly tagged with this concept), but also lots of other sights. Second, YouTube tags do not necessarily indicate that an object is visually present: an example for this is a portrait showing the constructor of the Eiffel Tower, Gustave Eiffel: while the tag "Eiffel Tower" is appropriate to a certain user with specific background knowledge and expectations, the object "Eiffel Tower" is not visible, and correspondingly to many users the concept is absent.

An illustration of typical web video content for the concept "basketball" is given in Figure 4.1. It can be seen that — while the concept is present in some frames — others are not visually related to it at all. We will refer to the first kind of frames as *relevant*, while calling the latter *non-relevant*. Overall, we will refer to web video as a *weakly labeled* data source with significant *label noise*, meaning that the associated tags are only weak indicators of concept presence.

Let us consider what influence label noise has on concept detector training. Typically, for each target concept a binary classification problem is formulated of differentiating concept presence from concept absence. Correctly and accurately labeled training content is assumed to be given: for example, only the top frames in Figure 4.1 are used as positive training samples for the concept "basketball". However, in case of label noise, positive training samples also contain non-relevant content (like the bottom frames in Figure 4.1). If using standard supervised learning, the resulting detectors tend to detect content similar to this non-relevant content. Consequently, it is to be expected (and will be demonstrated later) that concept detection performance degrades with an increasing amount of non-relevant training content.

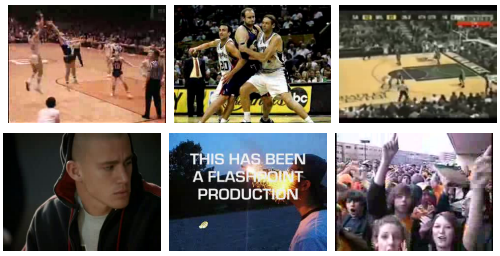


Figure 4.1: Keyframes from YouTube clips tagged with "basketball". While some frames do show basketball (top), other *non-relevant* content is not visually related to the concept (bottom). Pictures from YouTube.

While several studies regarding web *image* content have been presented previously (they will be discussed together with other related work in Section 4.2), the work presented here is the first one addressing weak labels for the video domain, with experiments conducted on a YouTube-based dataset of $22,000$ web video clips. We will first see that significant amounts of noise material (typically, $50-80\%$) have to be expected, and that this degrades the performance of standard detectors severely (Section 4.3).

This opens the question whether the robustness of concept detection with respect to label noise can be improved. For this purpose, an approach called *relevance filtering* is proposed. This method is based on a extension of the statistical models underlying concept detection such that non-relevant material is identified and filtered in a process that interleaves outlier elimination and model learning. Note that this approach has parallels with relevance *feedback* in information retrieval [RL03]: in both cases, we want to identify and discard content that is non-relevant for a query (in case of relevance feedback) or for a concept (in case of relevance filtering). However, while relevance feedback is used for a reranking at retrieval time, the filtering proposed here is coupled with system training, and the ultimate goal is an improved accuracy of the resulting concept detectors.

It will be shown that relevance filtering can be used as a wrapper around well-known supervised learning techniques, for generative models (kernel density estimation) as well as for discriminative ones (Support Vector Machines). In quantitative experiments (Section 4.5), it is shown that — without any manual supervision — relevance filtering is capable of identifying relevant content and filtering out non-relevant one. Further, detectors trained with relevance filtering are demonstrated to outperform their supervised counterparts.

4.2 Related Work

In this section, research related to visual learning under label noise is outlined. A review of conventional supervised concept detection (which can be found in Chapter 2) will be omitted here, and instead the focus will be on the aspect of weak label information. This setup is viewed from two different perspectives.

The first view is domain independent: it addresses the problem of weak labels from a machine learning perspective and proposes statistical models and learning algorithms that require less information compared to fully supervised techniques. These methods have been subsumed under the term *semi-supervised* learning (Section 4.2.1). The second perspective is domain-specific and addresses learning from weakly labeled image and video content. One setup in this field is learning from web images, a challenge that has started to attract researchers' attention over the last years. (Section 4.2.2). Also, a few contributions have been made for weakly labeled videos, which will be outlined in Section 4.2.3.

4.2.1 Semi-supervised Learning

This section discusses semi-supervised learning, a class of machine learning techniques for dealing with incomplete label information. One frequently studied setup in this area [CSZ06] assumes a (usually small) set of samples $X_L = \{x_1, ..., x_l\}$ with class labels $y_1, ..., y_l$ to be given. A second (usually large) set of samples $X_U = \{x_{l+1}, ..., x_N\}$ is available as well, but the associated labels are unknown (or *latent*). This setup can be seen as a borderline case between supervised learning (where all training data is labeled, i.e. $X_U = \varnothing$) and unsupervised learning (where no labels are given at all, i.e. $X_L = \varnothing$). While supervised methods can only use X_L, semi-supervised learning can exploit X_U as well (which can be viewed as evidence on the sample distribution $p(x)$).

To do so, a variety of strategies has been proposed [CSZ06, Zhu05]. One approach is to infer labels for samples in X_U and then treat them in a fully supervised framework. In its simplest form, this *self-training* approach is an iterative wrapper around a base classifier, in which samples are iteratively classified and the training set is expanded with a selection of the newly labeled data (usually the ones for which the classifier is most confident). As an extension, *co-training* [BM98] has been suggested, where multiple classifiers are trained on different features and "teach" each other.

Another approach formulates semi-supervised learning as a parameter estimation problem under missing data. A marginalization over latent class labels is

done, and parameters θ are estimated by maximizing the likelihood. For optimization, the *Expectation Maximization* (EM) algorithm [Del02, DLR77] is used, which alternates two steps in a local search: first, label posteriors $P(y_i|x_i, \theta_t)$ for samples in X_U are estimates from the current parameter estimate θ_t ("E"-step). Based on this information, the system parameters are updated to a new version θ_{t+1} ("M"-Step).

Other techniques include Transductive Support Vector Machines [Joa99], which integrate unlabeled samples in a margin maximization framework, and graph-based methods, which propagate label information throughout a graph with samples as nodes and similarity-weighted edges [CSZ06, Ch. 11].

While the above formulation of semi-supervised learning has experienced a boost of research over the last years, a similar setup called *adaptation* has been studied even earlier in the domain of optical character recognition (OCR). Imagine a multi-font character recognizer being applied to a test document. This setup resembles semi-supervised learning in the sense that labeled training samples are complemented with unlabeled ones, namely letters from the test document. As the distribution of this target set may be different from the training data (for example, showing a specific font), it seems reasonable to make use of the unlabeled data, i.e. to *adapt* the classifier. This makes the problem similar to the semi-supervised learning formulation above, and in fact related methods have been proposed: for example, Baird and Nagy [BN94] follow a self-training approach: a base classifier trained on multiple fonts is applied to the target document, and classification results are used as labeled samples for additional training iterations.

Another adaptation approach is based on a classification by clustering [Bre01a, Bre01b]: characters in the test document are clustered to coherent groups, which are then reliably mapped to class labels using results of a multi-font base classifier [Bre01b]. The resulting decision boundaries coincide with cluster boundaries — i.e., like in semi-supervised learning unlabeled samples are employed to adapt the classifier.

A look at work on OCR adaptation also makes clear that the understanding of semi-supervised learning should not necessarily be limited to the presence or absence of labels. Instead, other kinds of information may be exploited to improve recognition further: some of these have already been pointed out for the video domain, like segmentation information (Section 2.4). In the case of OCR, test samples share certain characteristics (for example, letters within a document usually show consistent fonts and degradations). This is referred to as *style*, and adaptation can exploit *style coherence* as an additional information source. One

such approach has been proposed by Sarkar and Nagy [SN05]. It models the style of the test document as a latent random variable, which is inferred using a maximum-likelihood approach. Then, a pre-trained style-specific classifier is used for an accurate recognition. Another model has been proposed by Mathis and Breuel [MB02] based on Hierarchical Bayesian methods, where style is modeled as a continuous parameter guiding the sample generation process.

Style modeling will be of interest in the context of concept detection in Chapter 5 of this thesis. Correspondingly, an in-depth discussion of the topic will be omitted here and provided later instead (Section 5.2). For now, it should only be kept in mind that semi-supervised learning is not necessarily limited to learning from a partially labeled training set, as is demonstrated by work on OCR adaptation.

4.2.2 Web Images

Visual learning from web image content, which can be acquired via text-based search engines as offered by Google or Yahoo!, has entered the focus of computer vision research over the last years. The data source is similar to web video in the sense that label information is weak and that large parts of the retrieved training data may be junk: for example, a text-based query for "airplane" to Google or Yahoo! image search also returns content not visually related to airplanes. Fergus et al. [FFFPZ05] have reported label precisions between 18% and 77% for Google Image Search, Schroff et al. [SCZ07] an average precision of 39%.

A variety of approaches has been suggested for acquiring sets of training images for object recognition from the web [BF06, SCZ07, SSTK08, YB05]. These methods are targeted at a content-based refinement of raw image sets obtained by text-based search. Usually, a three-step procedure is applied. First, a raw set of images is downloaded. Second, a subset of "good" candidate images for concept presence is selected, which can be done using manual annotation [BF06] or an analysis of text and meta-data surrounding the image [SCZ07, YB05]. Finally, a statistical model of concept presence (a Support Vector Machine [SCZ07], a region-level annotation model [BDF$^+$03], or a mining procedure based on a heuristic saliency measure [SSTK08]) is trained on the refined image set and used to re-rank the web content. These approaches are related to the work in this chapter as they are targeted at a refinement of training sets. Yet, they are limited in the sense that they do not cover the actual learning of concept models (though this happens inherently in some filtering approaches). Instead, this chapter is targeted at a joint training set refinement and model learning, and the focus is on the performance of the resulting detectors.

Other related work follows a more similar approach to the one presented here and combines training data refinement with model learning. Fergus et al. [FFFPZ05] learn visual models of object categories from Google's image search. The method uses a topic model (or extensions that take the spatial arrangement of local patches in the image into account) to cluster an image collection. The idea behind this is that images showing the target object accumulate in a single *relevant* cluster, which is then used for classification. The system has been tested by training on web images and applying the resulting classifiers to standard benchmarks, where results are significantly better than random guessing but do not reach the accuracy of training on the target domain. A clear limitation of the approach is the assumption of a single relevant topic.

In contrast to Fergus' approach [FFFPZ05], the OPTIMOL system by Li et al. [LWFF07] follows an incremental strategy, i.e. a training set is agglomerated and an object model is learned in parallel. The approach works in a self-training fashion, starting from an initial, highly accurate set of sample images. Iteratively, a topic model is trained, and the pool of training data is expanded using a Bayesian decision. The approach has been demonstrated to outperform Fergus' system. Yet, a problem remains in the initialization with *good* training samples, which has been reported to be a crucial factor for a similar system [MPN08].

Finally, another approach is to perform a filtering similar to the one presented in this chapter [WS08, LSW08]. A nearest neighbor analysis of training images is done, and images or tags are rejected if they are "strange", i.e. if no nearest neighbors with similar tags are found. Using the resulting filtered training sets, improved recognition performance is demonstrated on the Caltech-101 benchmark and on web-based training sets.

4.2.3 Weakly Labeled Videos

Only a few contributions in the literature address the fact that video data may come with label noise. Gargi and Yagnik [GY08] point out that label information in videos may be *coarse*, which they refer to as the *label resolution problem*. They rely on a feature selection using AdaBoost to achieve a higher robustness. Gu et al. [GMH+07] cast concept detection as a multiple instance problem and propose to adapt the kernel function in an SVM framework. Both methods, however, correspond to mere feature fusions and do not model label weakness explicitly.

A contribution closer to the one presented here has been made by Wang et al. [WHS+06], who study concept detection in a semi-supervised setup (where only a few labeled samples are given initially). A kernel density model is extended

such that the contribution of each training sample is weighted by its class posterior, and an iterative fitting algorithm is proposed to match unlabeled content to classes. Performance improvements over supervised learning from only a few labeled samples are demonstrated. A similar model will be a part of the study presented in this chapter (Section 4.4.2), but beyond this the framework proposed here is also integrated with other, discriminative models.

Overall, the effects of label noise — regarding label precision, its effects on concept detectors, and ways to overcome it — are not fully understood, though a variety of promising techniques for filtering non-relevant content has been proposed. In this chapter, we will build on these techniques and investigate in detail how and to which extent they can help to overcome the label noise problem.

4.3 Experiments using Standard Methods

In previous sections, we have introduced web video as a *weakly labeled* information source for concept detector training. It has been outlined that significant label noise is to be expected, as the tags coming with web video are coarse, subjective, and context-dependent. Correspondingly, significant amounts of content are to to be considered *non-relevant* with respect to target concepts to be learned. This assumption will be validated in Section 4.3.1. Also, we will study what effect label noise has on concept learning when using standard techniques, and it will be shown that the performance of concept detectors degrades significantly (Section 4.3.2).

4.3.1 Label Noise in Web Tags

While standard training sets for supervised learning are assumed to come with accurate labels, positive training samples in web video datasets contain only a certain fraction of relevant samples. This *relevance fraction* is denoted with α in the following:

$$\alpha := \frac{\text{number of training samples showing the target concept}}{\text{number of all training samples labeled with the target concept}}$$

α can be seen as a measure of label noise. For accurate annotations, we expect it to be close to 100%. For web video, it is unknown a priori what percentage of training material is in fact related to a target concept. Further, this fraction may differ between concepts: while for some concepts high-quality training sets may be obtained, others may be used as tags often but *appear* only infrequently.

To get a deeper insight into the label noise of YouTube-based training material, a manual assessment for 10 test concepts was done ("basketball", "beach", "cats", "desert", "eiffeltower", "helicopter", "sailing", "soccer", "swimming", and "tank"). These concepts were chosen from the YouTube-22concepts dataset used in Chapter 3 with respect to a good coverage of concepts, including various objects, locations, and sports. For each concept, a canonical definition was formulated (Appendix A). These definitions were chosen such that the concept should be directly visible and recognizable, and no particular context or prior knowledge should be required. For example, the concept "basketball" applies to scenes showing a basketball or streetball game, but not to scenes of a cheering crowd. YouTube material was manually assessed according to these definitions. For each of the 10 test concepts, 400 clips were acquired from YouTube, and only the first 7 Megabytes (ca. 2.5 minutes) per video were used. The motivation for this was that material within a clip can be strongly redundant, such that a higher diversity of material can be achieved by using *small* samples from *many* videos. For the same reason, we did not download more than a single video per YouTube user (users often produce series of similar clips, which would have reduced diversity significantly). The overall length of the dataset is about 100 hours.

Two kinds of queries were used to download videos from YouTube:

1. **Raw Queries**: The query only consists of a single tag describing the concept, like "beach". This corresponds to a fully automatic setup, in which a concept detection system is just given a vocabulary of tags and YouTube is crawled fully automatically for training material.

2. **Refined Queries**: Querying the YouTube API with a single tag must be expected to give very noisy results. For example, the query "beach" does not only return scenes of beaches, but also music videos by the "Beach Boys" and parties in Daytona Beach City. While these may be valid annotations to the video owner, they must be considered distracting when it comes to learning a visual concept (like "scenes showing a beach"). To improve the quality of downloaded material, two refinements were made. First, the fact was used that videos at YouTube are organized in categories like "Pets&Animals" or "Autos&Vehicles". The download was restricted to a canonical category (like "Travel&Places" for "beach", which excludes most music videos). Second, queries were refined according to a brief analysis of the first YouTube results page. For example, the query "beach" was replaced with "walk on the beach", which filtered out city names. The exact list of final queries is given in Appendix A.

Table 4.1: A manual annotation of training material downloaded from YouTube shows that significant label noise occurs. Only a certain fraction α of web video training sets (in most cases below 50%) does in fact show the target concept.

fraction of relevant training content α (%)					
concept	raw query	refined	concept	raw query	refined
basketball	20.5	40.6	helicopter	14.6	38.1
beach	15.6	44.3	sailing	16.4	26.2
cats	47.6	50.1	soccer	25.3	43.7
desert	11.4	19.0	swimming	23.4	60.0
eiffeltower	21.4	39.7	tank	14.5	24.3
			average	**21.1**	**38.6**

For both kinds of queries, material was downloaded and keyframes were extracted using a simple change detection: frames were scaled to 28×21 pixels, and a new keyframe was sampled when the sum of its pixel differences to the last keyframe exceeded a manually set threshold $t = 49$. To control the number of keyframes, the minimum time interval allowed between two successive keyframes was set to 4 seconds and the maximal one to 300. Manual assessments were done for at least $1,000$ frames per concept, with an overall number of $89,500$ annotations.

Results of the annotation process are given in Table 4.1. They indicate that significant label noise is to be expected when learning concepts from YouTube clips. The downloaded content contains significant fractions (in most cases more than 50%) of non-relevant material. This is not necessarily because tags are incorrect: often, there is a subtle connection between the content and a given tag. For example, it is reasonable to a specific user to label a report about Gustave Eiffel with "Eiffel Tower". Yet, according to the target concept definition ("scenes showing the Eiffel Tower"), the concept is not visually present in this scene. It can be seen that α is particularly low for raw queries (21.1% on average), and that a manual query refinement leads to better results (38.6%). Further, the percentage of relevant material varies between concepts: for example, the fraction of material found to be relevant varies between 19.0% ("desert") and 60% ("swimming") for refined queries.

These results confirm similar observations made previously for the image domain: for datasets based on image search, precisions of 39% have been reported for object category recognition [SCZ07]. For Flickr images, Kennedy et al. [KCK06]

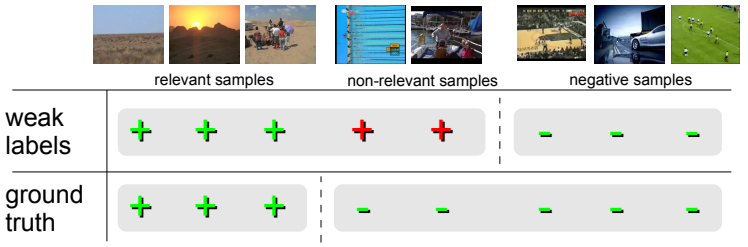


Figure 4.2: Sampling a training set for the concept "desert" and $\alpha = 60\%$. Non-desert content models the background class (right). Positive samples are mixed of 60% desert frames and 40% non-desert frames (which are incorrectly labeled as relevant). This weakly labeled setup (top) is compared with learning from correct labels (bottom).

have observed an accuracy of 50% for New York sights. Since for video data the *coarseness* of labels is an additional problem, it may seem surprising that the quality of YouTube material is not much lower compared to image datasets. Possible explanations for this are that video contains less graphical material (as logos or clip arts, which are typically found in web search results) and shows more "regular" content (many Flickr images are to be considered artwork, for which the connection to a tag can be very subtle).

4.3.2 How Label Noise Affects Concept Learning

In the last section, significant label noise for YouTube content has been reported. The next question is how this influences concept detection if using standard methods. Usually, the machine learning techniques underlying concept detection consider all positively labeled training content to truly show the concept. Intuitively, it can be expected that this approach — if trained on weakly labeled content — will lead to an inaccurate detection.

In this experiment, we validate this hypothesis and quantify performance degradation. The annotations described in the last section serve as ground truth labels. According to them, content is randomly compiled into training sets of varying relevance fraction α, and the resulting concept detectors are evaluated on a held-out test set.

Setup The same 10 test concepts, frames, and annotations as in Section 4.3.1 were used. For each frame, a bag-of-visual-words feature (see Section 3.5) was

extracted by a grid sampling of ca. $3,600$ patches at several scales, which were described with SIFT features [Low99]. These were matched with a $2,000$-dimensional visual codebook learned previously on a large dataset of 81 concepts. Finally, dimensionality reduction was applied using PLSA [Hof01], obtaining a 64-dimensional feature vector per frame. This dimensionality reduction was done for efficiency purposes and has previously been validated to give comparable results to the full visual word histograms.

Random datasets were sampled for different levels of label noise $\alpha \in \{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.7,\ 1.0\}$ (as typical values for web video lie between 20% and 50%, a stronger focus was put on this range). This sampling is illustrated for the concept "desert" and $\alpha = 60\%$ in Figure 4.2: negative samples — which can be obtained easily from videos not tagged with the concept — represent the background class. Positive samples consist of 60% true positives (which are manually assessed to show the target concept) and 40% non-relevant frames (*false positives*), which are again drawn randomly from YouTube videos *not* tagged with the concept. Further, correctly annotated test sets were sampled:

for $\alpha \in \{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.7,\ 1.0\}$:

1. *sample training set*
   - sample $1,000$ non-relevant frames with label -1
   - sample $\alpha \cdot 500$ relevant frames with label 1 ("true positives")
   - sample $(1-\alpha) \cdot 500$ non-relevant frames label 1 ("false positives")
2. *sample test set*
   - sample 500 relevant frames with label 1
   - sample $1,500$ non-relevant frames with label -1

It was made sure during sampling that no material from the same video clip was assigned to both training set and test set. Also, it should be noted that only the training set is weakly labeled, while the test set uses ground truth to assure a precise evaluation. Five random datasets were resampled, and results were averaged over these runs. As a performance measure, *average precision* was used, which is a standard choice for concept detector evaluation [KO07] and has already been used in Chapter 3. Tests were run for two standard supervised learning approaches: a generative model (kernel densities) [DHS00] and a discriminative one (Support Vector Machines) [SS01].

Kernel Densities: Given training samples $x_1, ..., x_n$ with labels $y_1, ..., y_n \in \{-1, 1\}$, kernel densities model class-conditional densities of concept presence, p^1,

and absence, p^0 (Z and Z' are normalization factors):

$$\begin{aligned} p^1(x) &= \frac{1}{Z} \sum_{i:y_i=1} K_h(x; x_i), \\ p^0(x) &= \frac{1}{Z'} \sum_{i:y_i=-1} K_h(x; x_i), \end{aligned} \tag{4.1}$$

and a test frame x is scored using Bayes' rule (the class prior — which does not influence the ranking of test items — is assumed to be uniform):

$$P(y = 1|x) = \frac{p^1(x)}{p^1(x) + p^0(x)}$$

As a kernel function, the well-known Epanechnikov kernel with Euclidean distance function is used:

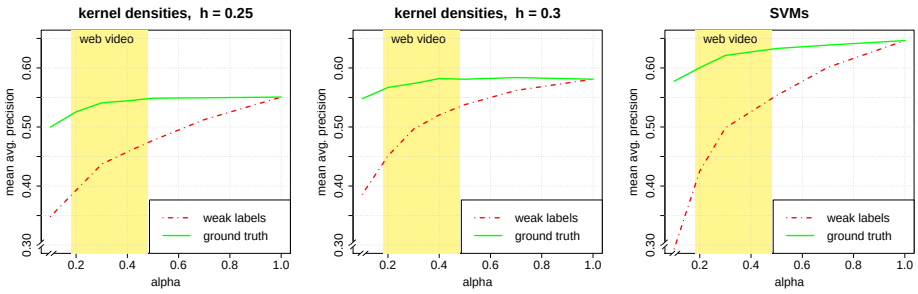
$$K_h(x; x') = \frac{3}{4} \cdot \left(1 - \frac{||x - x'||^2}{h^2}\right) \cdot 1_{(||x-x'|| \leq h)}$$

This choice was made for efficiency reasons: since the Epanechnikov kernel has local support, it can be combined with fast nearest neighbor search [PPC01]. The kernel bandwidth h is a free parameter of the system. It has been reported to have a strong influence on the resulting kernel densities [Tur93], as high values of h lead to a smoother density in feature space. For the experiment presented here, several choices of $h \in \{0.225,\ 0.25,\ 0.275,\ 0.3,\ 0.325\}$ were tested. The optimal performance was achieved for $h = 0.275$. However, to illustrate the effects of different bandwidths more clearly, results are reported for a low value ($h = 0.25$) and a high one ($h = 0.3$).

Support Vector Machines (SVMs): As a representative discriminative approach, SVMs [SS01] were tested. A Radial Basis Function (RBF) kernel was used, and the smoothness and cost parameters σ and C [SS01] were evaluated in a grid search maximizing the cross-validated mean average precision. For efficiency reasons, no complete search was done for each run, but the values $C = 5$ and $\sigma = 25$ were validated to give stable good results over different concepts. SVM scores were mapped to class posterior estimates using the standard approach by Wu et al. [WLW04].

Both systems — kernel densities and SVMs — were tested in two setups:

- **Weak Labels**: Only a fraction α of positive training samples does truly show the concept, which corresponds to the practical situation of learning from web video.



(a) (b) (c)

Figure 4.3: Comparing concept detection when trained on ground truth labels and on weak labels ($n = 100,000$). The mean average precision over all 10 test concepts is plotted against the fraction of truly relevant training samples α. The two results on the left represent the kernel density system for bandwidths 0.25 (a) and 0.3 (b), the result on the right is for SVMs (c).

- **Ground Truth**: All training samples are given their correct class label, indicating how concept detection might work if noise content was filtered out.

Results Quantitative results are illustrated in Figure 4.3, where the system performance on the test data is plotted against α. A first observation is that the influence of label noise on the ground truth control run is negligible, as performance remains almost constant when varying α. This is intuitively correct, since noise samples (which become more frequent with lower values of α) are assigned their correct negative labels. A performance decrease for low relevance fractions of $\alpha \approx 10\%$ can be attributed to a low absolute number of positive training samples.

In contrast to this, the system trained on weakly labeled data suffers a significant performance loss. In the absence of noise ($\alpha = 1$), both systems give the same performance (which is trivial, because no non-relevant content occurs). When decreasing α, the percentage of noise in the training set increases and system performance degrades. For example, for training sets with 70% non-relevant content ($\alpha = 0.3$) and a bandwidth of 0.25, kernel density estimation trained on weakly labeled data gives a performance of 43.7%, while training on the correct labels gives 54.1%. The more noise in the training data, the stronger is the gap between the weakly supervised run and the control run. This observation can

be made for the generative model (Figures 4.3(a) and 4.3(b)) as well as for the discriminative one (Figure 4.3(c)).

We can now match these results with the observations made in the last section, which indicated that typical relevance fractions of web video data are in the range of $20-50\%$. This range is highlighted in yellow in all plots. If we focus on this area, we see that performance differences between the oracle-based control run and the standard detector range from 4% to 19%. Over the whole range, this difference is significant (sign test over rank improvement of positive samples, level 99%). This indicates that concept detection could be improved distinctly if we were able to filter out non-relevant content in the training set. This strategy will be followed by an approach called *relevance filtering* proposed in this chapter.

Finally, another interesting observation is that — when comparing the two bandwidths $h = 0.25$ and $h = 0.3$ — for the higher bandwidth, the difference between the standard detector and the control run is lower. For example, for a relevance prior of 50%, the difference is 7.1% for $h = 0.25$ but only 4.3% for $h = 0.3$. Obviously, a simple way to increase system robustness with respect to non-relevant training samples is to increase the bandwidth (which leads to smoother density representations in feature space). This draws an analogy to well-known techniques in signal processing, where smoothing filters are a common way to reduce outlier influence [GW02, Ch.5]. Note, however, that this overcomes label weakness only to a limited extent. The best overall performance is achieved for a moderate bandwidth of $h = 0.275$, which gives an appropriate balance between smoothing and relevance filtering. When increasing the bandwidth further, system performance degrades.

4.4 Relevance Filtering

In this section, the question is addressed whether the influence of label noise on concept learning can be reduced. To achieve such robustness, a novel approach is presented that explicitly takes non-relevant training material into account. This material is identified and filtered during concept detector training.

The approach will be referred to as *relevance filtering*. It is explicitly designed to cope with weakly labeled training videos, i.e. content annotated with a target concept is assumed likely to show the concept, but does not necessarily do so. The key idea is that relevant content appears frequently and forms clusters in feature space, while non-relevant material comes as outliers that can be identified and relabeled. This approach can be combined with a variety of well-known techniques

Table 4.2: An overview of the basic notation and concepts used in Section 4.4.

x	feature vector representing a test keyframe
$y \in \{-1, 1\}$	absence/presence of target concept in x
$P(y = 1\|x)$	keyframe score (to be estimated)
$x_1, ..., x_n$	feature vectors representing training frames
$\tilde{y}_1, ..., \tilde{y}_n \in \{-1, 1\}$	weak labels of concept presence in training frames (observed)
$y_1, ..., y_n \in \{-1, 1\}$	actual absence/presence of target concept in training frames (unknown)
$\beta_i := P(y_i = 1\|x_i, \tilde{y}_i)$	*relevance score*: the probability of a training frame to be relevant
$\alpha := P(y_i = 1\|\tilde{y}_i = 1)$	*relevance prior*: fraction of actually relevant training frames among potentially relevant ones (given)

from supervised learning. Two such extensions are presented in detail, one for a generative approach (kernel density estimation) and one for a discriminative technique (Support Vector Machines).

4.4.1 Basic Concepts

A video is represented via keyframes, and concept detection is effectively conducted on keyframe level in the following. Each frame is associated with a feature vector $x \in \mathbb{R}^d$. The presence of the target concept is denoted with a label y such that $y = 1$ indicates concept presence and $y = -1$ concept absence. The goal of concept detection is to estimate a *score* $P(y = 1|x)$. Training data is represented by keyframes (or associated features) $x_1, ..., x_n \in \mathbb{R}^d$. For each training frame x_i, an indicator of concept presence is given that tells us whether the concept *may* appear in the frame. This information is derived from user-generated tags in practice, and is denoted with a *weak label* $\tilde{y}_i \in \{-1, 1\}$. The *actual* presence of the target concept, however, is *latent*. It is denoted with $y_i \in \{-1, 1\}$. For each concept, a binary classification problem is cast according to the following definition:

> **Definition: *Weakly Labeled Classification Problem***
> Given training data in form of samples $x_1, ..., x_n \in \mathbb{R}^d$ with labels $\tilde{y}_1, ..., \tilde{y}_n \in \{-1, 1\}$, learn a scoring function $\phi : \mathbb{R}^d \to [0, 1]$ such that $\phi(x) \approx P(y = 1|x)$. Thereby, training labels are assumed to be weak indicators of true labels $y_1, ..., y_n$ such that:

- If the weak label is negative ($\tilde{y}_i = -1$), the true label is negative as well ($y_i = -1$).
- If the weak label is positive ($\tilde{y}_i = 1$), the sample *may* belong to the positive class, but does not necessarily do so, i.e. the true label y_i is unknown.

Further, a prior for weakly labeled samples to be truly positive is assumed to be given. It is denoted with $\alpha := P(y_i = 1|\tilde{y}_i = 1)$.

The key characteristic of this formulation is that true latent class labels are separated from given ones. The setup is regulated by the relevance fraction α, which tells us how many of the positively labeled samples do show the target concept. In practice, this information is not available, as it requires knowledge of the latent true labels y_i (we will discuss options to address this problem later).

Finally, it should also be noted that — while we model false positives (i.e. it is possible that $\tilde{y}_i = 1$ and $y_i = -1$) — false negatives ($\tilde{y}_i = -1$ and $y_i = 1$) are not taken into account. This means that if a frame is not labeled with the target concept, the concept is assumed to be absent. Strictly speaking, this is not true (for example, there might be web videos clips showing basketball that the user has simply forgotten to label with the concept). According to observations we made on web video, however, the amount of these false negatives is negligible.

Let us compare the weakly labeled classification problem above with other learning setups. When compared with standard *supervised learning*, two key differences can be observed: first, labels for the positive class are only weak indicators of the true labels. Second, an additional assumption is made (in form of α) on how much of the weakly labeled material does in fact show the target concept.

Compared with the *semi-supervised learning* setup as defined by Chapelle et al. [CSZ06], the above definition can be seen as a degenerate special case: the weakly labeled samples $\{x_i : \tilde{y}_i = 1\}$ can be viewed as *unlabeled*, as their true label y_i is not known. This leads to an extremely *imbalanced* problem: while semi-supervised learning usually assumes a few labeled samples of either class to be given, in our setup we are confronted with many samples from class -1 (simply because content not labeled with a concept can be obtained easily) but no sample of class 1 (since indicators of concept presence are weak). This renders a straightforward application of many semi-supervised algorithms impossible.

Finally, the weakly labeled learning setup strongly resembles the visual learning from noisy image sources like Google's image search [FFFPZ05, LWFF07, WS08].

The work in this chapter follows a similar strategy to these approaches (particularly to the ones by Wnuk and Soatto [WS08] and Li et al. [LSW08], who also propose a distribution-based filtering of training sets). Yet, several differences remain. First (and obviously), the web video domain addressed here differs from images delivered by web search engines. Video comes with an additional temporal structure, and the models proposed here take this structure into account (Section 4.4.4). Second (and more importantly), we do not cover a single statistical model, but view relevance filtering as a wrapper than can be applied around a variety of supervised learning techniques. For both a generative and a discriminative base model, relevance filtering extensions will be presented in the following.

4.4.2 Generative Case: Kernel Density Estimation

In this section, a first relevance filtering extension of a generative base model is presented, namely *kernel density estimation* [DHS00]. The relevance of training content is modeled as a latent random variable that is inferred during the learning procedure.

Class-conditional Densities and Scoring

Class-conditional densities of relevant and non-relevant content are modeled by the following weighted kernel densities p^1_β (concept presence) and p^0_β (concept absence):

$$\begin{aligned} p^1_\beta(x) &= \frac{1}{Z} \cdot \sum_{i=1}^{n} \beta_i \cdot K_h(x; x_i) \\ p^0_\beta(x) &= \frac{1}{Z'} \cdot \sum_{i=1}^{n} (1 - \beta_i) \cdot K_h(x; x_i), \end{aligned} \tag{4.2}$$

where $Z = \sum_i \beta_i$ and $Z' = n - Z$ are normalization constants. Compared to the fully supervised setup from Equation (4.1), the key difference is that p^1 and p^0 are now parameterized by a vector $\beta = (\beta_1, ..., \beta_n)$. This vector consists of *relevance scores* $\beta_i := P(y_i = 1 | \tilde{y}_i, x_i)$, which means that each training sample is weighted according to its probability of being relevant: if a training sample is likely to truly show the concept, it has a strong influence on the distribution of relevant samples p^1_β but low influence on p^0_β. This way, the uncertainty of label information is taken into account (a similar model has been used in a semi-supervised setup before [WHS+06]). Note that if we set the relevance scores according to the weak

labels:

$$\beta_i = \begin{cases} 1, & \tilde{y}_i = 1 \\ 0, & \tilde{y}_i = -1, \end{cases}$$

the system degenerates to the standard supervised case (Equation (4.1)), in which all positive samples are assumed to be relevant. This approach has already been tested in Section 4.3 and will again serve as a baseline in later experiments.

Training

To compute the class-conditional densities p^1_β and p^0_β, the vector of relevance scores β must be inferred in system training, which takes features $x_1, ..., x_n$, weak labels $\tilde{y}_1, ..., \tilde{y}_n$, and the relevance prior α as input. For each training frame x_i, the true label y_i is to be estimated (or more precisely, the associated probability β_i). Three situations may occur:

1. $\tilde{y}_i = -1$ (*negative sample*): If a frame x_i is *not* labeled with the concept, it is assumed to be non-relevant, i.e. $\beta_i = 0$.

2. $\tilde{y}_i = y_i = 1$ (*true positive*): A frame x_i is labeled with the concept and is in fact relevant. Accordingly, β_i should be high.

3. $\tilde{y}_i = 1, y_i = -1$ (*false positive*): A frame is labeled with the concept but is not relevant. Such samples appear due to label noise. For them, β_i should be low.

Let us assume that p training samples are weakly labeled with the concept, and that training samples are sorted such that $\tilde{y}_1 = .. = \tilde{y}_p = 1$ and $\tilde{y}_{p+1} = .. = \tilde{y}_n = -1$. While we know that $\beta_{p+1} = .. = \beta_n = 0$, the relevance scores $\beta_1, ., , \beta_p$ need to be estimated, i.e. training must divide potentially relevant frames into actually relevant ones and non-relevant ones. The key idea is to make this decision based on the distribution of training features: relevant content is assumed to cluster in certain regions of feature space, while outliers or samples close to negative content are considered non-relevant. Based on this assumption, two training procedures are suggested, a simple fixpoint iteration scheme and a Maximum a posteriori parameter estimation using Markov Chain Monte Carlo (MCMC) optimization.

Training 1: Fixpoint Iteration

In the following, the parameter vector β is restricted to the non-zero entries $\beta = (\beta_1, ..., \beta_p)$ (we know that $\beta_{p+1} = .. = \beta_n = 0$). One strategy to estimate β

is based on a fixpoint iteration in parameter space. First, relevance scores are initialized with the relevance prior: $\beta^0 = (\alpha, ..., \alpha)$. Then, the parameter vector β^k is iteratively updated to a new version β^{k+1} by plugging the current parameter estimate β^k into the class-conditional densities $p^1_{\beta^k}$ and $p^0_{\beta^k}$ (Equation (4.2)). From these densities, new estimates of relevance scores can be obtained using Bayes' rule:

$$\begin{aligned}
\beta_i^{k+1} &:= P(y_i = 1 | x_i, \tilde{y}_i = 1) \\
&= \frac{p(y_i = 1, x_i | \tilde{y}_i = 1)}{p(y_i = 1, x_i | \tilde{y}_i = 1) + p(y_i = -1, x_i | \tilde{y}_i = 1)} \\
&\approx \frac{P(y_i = 1 | \tilde{y}_i = 1) \cdot p(x_i | y_i = 1)}{P(y_i = 1 | \tilde{y}_i = 1) \cdot p(x_i | y_i = 1) + P(y_i = -1 | \tilde{y}_i = 1) \cdot p(x_i | y_i = -1)} \\
&\approx \frac{\alpha \cdot p^1_{\beta^k}(x_i)}{\alpha \cdot p^1_{\beta^k}(x_i) + (1 - \alpha) \cdot p^0_{\beta^k}(x_i)}
\end{aligned}$$

This process is repeated for a fixed number of iterations. Intuitively, the algorithm identifies regions in feature space where positively labeled frames concentrate and assigns high relevance scores to them, while outliers similar to negative content are given low relevance scores. The approach also resembles the well-known Expectation Maximization (EM) scheme [DLR77], which maximizes the data likelihood by alternating so-called "E" steps (in which posteriors for latent variables are estimated) and "M" steps (in which the parameters are updated according to estimates in the "E" step). If we compare the EM scheme to the fixpoint iteration used here, the relevance scores β_i resemble posteriors for latent variables in the EM scenario (namely, the true labels y_i). However, since the parameters of the class-conditional densities are equal to the relevance scores β and the framework is non-parametric otherwise, no "M" step is required.

The approach is also similar to the procedure used by Wang et al. [WHS+06], but the system is constrained in a different way. While Wang et al. addressed a strictly semi-supervised setup — where initial reliable training samples for all classes are available — we cannot rely on such information in our weakly supervised setup. Instead, we constrain the system with a certain *prior* of expected relevant material α. It should also be noted that if we choose the relevance prior to be $\alpha = 1$, it follows that $\beta_1 = \beta_2 = .. = \beta_p = 1$, such that the model degenerates to the supervised case (Equation (4.2)).

Training 2: MAP Parameter Estimation using MCMC Sampling

An alternative strategy to estimate the parameter vector $\beta = (\beta_1, ..., \beta_p)$ follows *Maximum a posteriori* (MAP) parameter estimation [DHS00, Ch. 3], i.e. the parameter vector with maximum posterior probability given the training data is chosen. The relevance prior α becomes part of a prior term in the target function:

$$\begin{aligned} \hat{\beta} &= \arg\max_{\beta} \quad p(\beta \,|\, x_1, ..., x_n, \tilde{y}_1, ..., \tilde{y}_n, \alpha) \\ &= \arg\max_{\beta} \quad \underbrace{p(x_1, ..., x_n \,|\, \tilde{y}_1, ..., \tilde{y}_n, \beta) \cdot p(\beta \,|\, \tilde{y}_1, ..., \tilde{y}_n, \alpha)}_{Q(\beta)} \end{aligned} \tag{4.3}$$

The target function $Q(\beta)$ consists of a data likelihood term $p(x_1, ..., x_n \,|\, \tilde{y}_1, ..., \tilde{y}_n, \beta)$ and a prior term $p(\beta \,|\, \tilde{y}_1, ..., \tilde{y}_n, \alpha)$. To estimate the former, samples are assumed to be independent and identically distributed, and for each frame x_i a marginalization over the latent label y_i is performed:

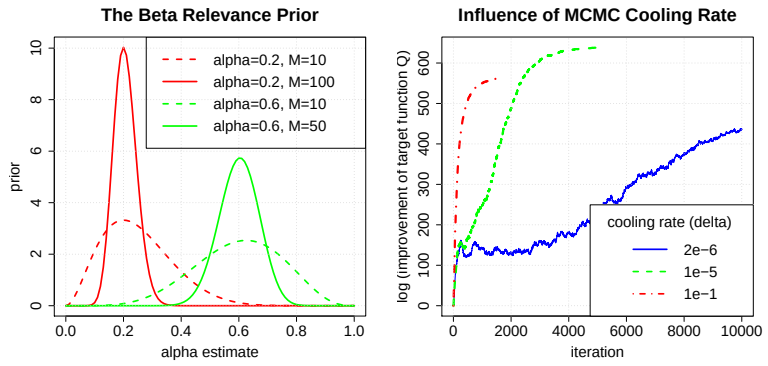
$$\begin{aligned} p(x_1, ..., x_n \,|\, \tilde{y}_1, ..., \tilde{y}_n, \beta) &\approx \prod_{i=1}^{n} p(x_i \,|\, \tilde{y}_i, \beta) \\ &= \prod_{i=1}^{n} [\, p(x_i, y_i = 1 \,|\, \tilde{y}_i, \beta) + p(x_i, y_i = -1 \,|\, \tilde{y}_i, \beta) \,] \\ &\approx \prod_{i=1}^{n} [\, p(x_i \,|\, y_i = 1, \beta) \cdot P(y_i = 1 \,|\, \tilde{y}_i, \beta) \\ &\qquad + p(x_i \,|\, y_i = -1, \beta) \cdot P(y_i = -1 \,|\, \tilde{y}_i, \beta) \,] \\ &= \prod_{i=1}^{n} [\, \tilde{\alpha} \cdot p^1_\beta(x_i) + (1 - \tilde{\alpha}) \cdot p^0_\beta(x_i) \,], \end{aligned}$$

where an estimate of the relevance prior from the relevance scores is used: $\tilde{\alpha} = \frac{1}{p} \sum_{i=1}^{p} \beta_i$.

The prior term $p(\beta \,|\, \tilde{y}_1, ..., \tilde{y}_n, \alpha)$ is chosen according to two criteria: first, the concept cannot be present in frames with negative labels, i.e. $\beta_{p+1} = .. = \beta_n = 0$. For the remaining relevance scores, we make use of the relevance fraction α, i.e. we enforce that $\tilde{\alpha} \approx \alpha$. Therefore, the prior $p(\beta \,|\, \tilde{y}_1, ..., \tilde{y}_n, \alpha)$ is modeled as a beta distribution:

$$p(\beta \,|\, \tilde{y}_1, ..., \tilde{y}_n, \alpha) = \frac{1}{Z} \cdot \tilde{\alpha}^{\alpha \cdot M} \cdot (1 - \tilde{\alpha})^{(1-\alpha) \cdot M} \tag{4.4}$$

where Z is a normalization constant. The parameter $M \in \mathbb{N}$ regulates the influence of the prior. A sample illustration with different choices of M and α can be found



(a) (b)

Figure 4.4: (a) The prior $p(\beta|\tilde{y}_1, ..., \tilde{y}_n, \alpha)$ is always peaked at α and thus biases the estimate $\tilde{\alpha}$ towards the true relevance prior α. (b) Simulated annealing for different choices of the cooling rate δ. While a slow cooling is inefficient and a greedy cooling gets caught in a suboptimal local maximum, a moderate cooling rate of $\delta = 10^{-5}$ offers a good tradeoff.

in Figure 4.4(a), which demonstrates that the prior is always peaked at α, and that this peak grows stronger with increasing M.

To optimize the target function Q from Equation (4.3), a stochastic *simulated annealing* procedure from the class of Markov-Chain Monte Carlo (MCMC) algorithms is used [AdFDJ03]. The general strategy of such optimizers is to draw samples from the target function using random walks. If applied to the relevance estimation scenario studied here, we obtain the following procedure: starting from an initial parameter vector $\beta^0 = (\alpha, ..., \alpha)$, a random walk $\beta^0, \beta^1, ...$ in parameter space is generated. A new parameter vector β^{k+1} is computed from β^k in two steps: first, an randomized update function q suggests a new version $\beta^* = q(\beta^k)$. If β^* improves Q compared to β^k, this change is accepted, i.e. $\beta^{k+1} = \beta^*$. Otherwise, β^* is only accepted with a probability of $\frac{Q^{1/T}(\beta^*)}{Q^{1/T}(\beta^k)}$, and in case of rejection the old version is kept ($\beta^{k+1} = \beta^k$). A *temperature parameter* T influences the strictness of rejection. This approach can be demonstrated to optimize the target function Q [AdFDJ03].

Design choices need to be made regarding the update strategy q and temperature T. Here, q is kept very simple: two random training samples $1 \leq j_1 < j_2 \leq p$ are picked, and a random amount of relevance mass $\epsilon \sim \mathcal{U}_{[0,0.1]}$ is shifted from one

sample to the other:

$$q(\beta^k) = (\beta_1^k,\ ..,\ \beta_{j_1}^k - \epsilon,\ ..,\ \beta_{j_2}^k + \epsilon,\ ..,\ \beta_n^k)$$

All other relevance scores remain unchanged, such that the update does not affect $\tilde{\alpha}$ and can be computed efficiently. To change the overall relevance fraction, a more time-consuming update step is conducted occasionally (every 250 iterations) that changes all relevance scores simultaneously according to a change coefficient $\gamma \sim \mathcal{U}_{[0.8,1.2]}$:

$$q(\beta^k) = (\gamma \cdot \beta_1^k, ..., \gamma \cdot \beta_n^k)$$

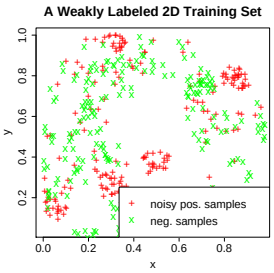
A sensitive parameter is the temperature T. The lower T, the stricter updates are rejected that do not improve Q, and the greedier optimization approaches a local maximum. According to standard practice [AdFDJ03], we choose the temperature T to decrease with increasing iterations at a "cooling rate" δ: $T = e^{-\delta \cdot k}$. The effects of different choices of δ are illustrated in Figure 4.4 for a weakly labeled "soccer" training set. The improvement of the target function Q (relative to the starting value) is plotted over the number of training iterations for different choices of δ. For low cooling rates, optimization is inefficient and progresses slowly. For high cooling rates, optimization is greedy and converges to a suboptimal local maximum. A moderate cooling rate of 10^{-5} offers a good tradeoff and converges to a better solution in a reasonable number of iterations.

A Sample Problem

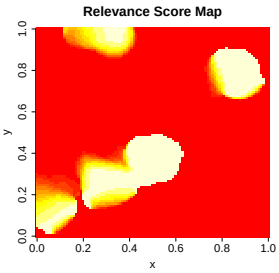
An illustration of relevance filtering for kernel densities is given in a small experiment. A 2-dimensional weakly labeled dataset is generated such that samples from the positive class contain a certain amount of incorrectly labeled false positives. For two classes (representing concept presence and absence), random prototypes are drawn from $[0, 1]^2$. For class 1, five prototypes are used, and for the negative class 50 prototypes. Samples are drawn from the surrounding of these prototypes according to kernel densities p^1 and p^0 with bandwidth $h = 0.05$. First, a training set of $n = 200$ noisy positive samples is generated:

$$x_1, ..., x_n \sim \alpha^* \cdot p^1 + (1 - \alpha^*) \cdot p^0.$$

The fraction of relevant samples is varied such that $\alpha^* \in \{0.2,\ 0.6,\ 1\}$, i.e. we use one clean training set ($\alpha^* = 1.0$), one with moderate label noise ($\alpha^* = 0.6$) and one with lots of non-relevant samples ($\alpha^* = 0.2$). For each training set, negative training samples drawn from $p^0(x)$ are added. A typical dataset used



(a)



(b)

error rate (%)			
α^*/α	0.2	0.6	1.0
0.2	**27.7**	32.2	35.5
0.6	13.0	**12.1**	15.8
1.0	11.0	10.9	**4.1**

(c)

Figure 4.5: (a) A 2D sample training set. Positive samples concentrate in 5 peaks, but contain 40% outliers. (b) A learned relevance map identifies relevant content near the correct five peaks. (c) Bag classification error rates for an experiment on synthetic data, whereas the true relevance fraction α^* and the assumed relevance prior α are varied. A choice of $\alpha = \alpha^*$ gives the best classification results ($n = 400,000$).

in this experiment is illustrated in Figure 4.5(a) — it can be seen that positive (red) samples concentrate near the five prototypes of class 1, but many outliers occur. Finally, a test set of equal size was sampled. This experiment was repeated 100 times, whereas for each run the relevance filtering framework was tested with fixpoint iteration training and a relevance prior of $\hat{\alpha} \in \{0.2, 0.6, 1.0\}$. Note that we distinguish between the true relevance fraction α^* and the relevance prior we expect, α.

The result of relevance filtering is illustrated in Figure 4.5(b), where a *relevance map* plots the relevance score β over feature space. It can be seen that high relevance scores are assigned to samples accumulating near the five prototypes, while outliers close to negative samples are identified as non-relevant. Classification results when applying the kernel density model with relevance filtering are reported in Table 4.5(c). Two observations can be made: first — and not surprisingly — the overall error rate of classification increases with the amount of noise material in the training set. The second observation is that the noise level α^* and the optimal choice of the relevance prior α are correlated, i.e. the lowest error rate is achieved for $\alpha = \alpha^*$. For example, for the clean training set ($\alpha^* = 1$) the supervised system ($\alpha = 1$) — which corresponds to a plain kernel density system assuming all positive training content to be relevant — performs best, while the weakly supervised systems ($\alpha = 0.2, 0.6$) incorrectly identify some content as non-relevant

Algorithm 1 Discriminative relevance filtering: samples are iteratively refined by training a base classifier, scoring training content, and relabeling the samples most likely to be false positives.

for $i = 1, ..., n$: set $\beta_i = 1$ (if $\tilde{y}_i = 1$) or $\beta_i = 0$ (otherwise).
randomly split $X = \{x_1, ., , .x_n\}$ into five folds $X_1, ..., X_5$
while $\frac{1}{p} \sum_{i=1}^{p} \beta_i > \alpha$ **do**
  for $k = 1, ..., 5$ **do**
    train a classifier on $X \setminus X_k$
    apply the classifier to X_k, obtaining *scores* σ
    for the N_f samples $x_i \in X_k$ with $\beta_i = 1$ and lowest scores $\sigma(x_i)$ **do**
      set $\beta_i = 0$
    end for
  end for
end while

and thus ignore valuable training information. On the other hand, if $\alpha^* = 0.2$, the best performance is achieved for $\alpha = 0.2$, and error is reduced by 7.8% compared to the supervised case. Generally, this result indicates that relevance filtering can improve classification on weakly labeled training sets.

4.4.3 Discriminative Case: Support Vector Machines

While a generative technique was integrated with relevance filtering in the last section, a similar extension will be presented for discriminative models in the following. The approach can be applied as a wrapper around a variety of base classifiers, with the only requirement that these deliver a class posterior estimate (or *score*, respectively). As a sample classifier, SVMs are used, which can be considered a standard choice for concept detection [vdSGS08, YCKH07, YH08].

The basic idea of relevance filtering for discriminative methods is similar to a semi-supervised self-training, but works in a filtering fashion instead of an incremental one. Iteratively, the base classifier is trained and used to identify potential false positives in the training set. Their relevance scores β_i are set from 1 to 0 (i.e., they count as negative samples in later iterations). This way, the weakly labeled positive samples are iteratively filtered and the model is refined. The whole process is repeated until the estimated relevance prior $\frac{1}{p} \sum_i \beta_i$ — which constantly decreases due to relabeling — reaches the expected relevance prior α. The whole training procedure is outlined in Algorithm 1 (note that filtering is done in a cross-

validation fashion to avoid overfitting). Also, an illustration is given in Figure 4.6: before relevance filtering, the decision boundary of the underlying classifier is suboptimal due to a false positive in the training set. This is identified based on its score during the learning procedure, and after relabeling it, a better decision boundary is achieved.

Let us compare the approach with the generative version from the last section. Generally, both approaches follow the same idea, namely to estimate the relevance of training content using the distribution in feature space and a relevance prior. However, two key differences can be identified. First, while the generative approach relies entirely on the distribution of content in feature space, the discriminative technique is inherently bound to the base classifier used. Second, the discriminative relevance filtering approach is not probabilistic. The scores used for filtering may be interpretable as relevance posteriors, but do not necessarily have to be. Also, no uncertainty is allowed regarding the relevance of a sample, but only a complete relabeling from the positive to the negative class takes place. Instead, the generative approach allows a soft assignment of samples to classes.

4.4.4 Temporal Neighborhood Suppression

As introduced so far, relevance filtering can be equally applied to image and video content (if represented by keyframes). In this section, an extension is presented that is specific to video content, which is known to come with an additional temporal structure. More precisely, the problem is addressed that video content appearing at about the same time (for example, keyframes sampled from the same shot) tends to be visually related. This can lead to inherent concentrations in feature space that compete with concentrations due to relevance. For example, imagine a very long YouTube clip showing an interview about basketball. Though the video is not visually related to the concept, it is marked as positive with a weak label. As the keyframes of the video are all similar, they form a cluster in feature space and boost each other to an incorrect high relevance score.

Obviously, the occurrence of such sample concentrations is related to keyframe selection, and one solution might be to enhance keyframe extraction such that clusters of keyframes are identified and boiled down to a single representative. On the downside, this comes with an inherent loss of information, and it is difficult to achieve cluster suppression while at the same time preserving a sufficient richness of content. Instead, a different strategy is outlined in the following that defines a *temporal neighborhood* $\mathcal{N}^T(x_i) \subset \{x_1, ..., x_n\}$ around each training sample x_i. For example, $N^T(x_i)$ might be all other training samples from the same YouTube

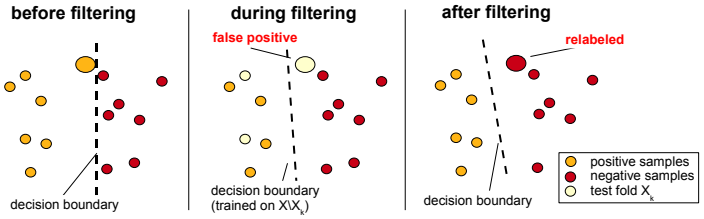


Figure 4.6: An illustration of discriminative relevance filtering on a dataset including a false positive. Before filtering, this sample degrades the model. During filtering, it is identified as potentially non-relevant material and relabeled, which leads to a better decision boundary.

clip as x_i. Then x_i should be marked relevant only if other potentially relevant content *outside* this temporal neighborhood is found to support x_i's relevance. This approach is referred to as *temporal neighborhood suppression* (TNS) in the following, and it is described for both the generative and the discriminative version of relevance filtering.

Generative TNS Temporal neighborhood suppression in the generative setting can be achieved by a slight modification. Basically, whenever the class-conditional densities (Equation (4.2)) are computed during training, content from the temporal neighborhood is skipped.

$$p^1_\beta(x_i) = \frac{1}{Z(i)} \cdot \sum_{x_j \notin \mathcal{N}^T(x_i)} \beta_j \cdot K_h(x_i; x_j),$$

$$p^0_\beta(x_i) = \frac{1}{Z'(i)} \cdot \sum_{x_j \notin \mathcal{N}^T(x_i)} (1 - \beta_j) \cdot K_h(x_i; x_j),$$

Note that this also requires a change of the normalization factors Z and Z', which now depend on the training sample x_i.

Discriminative TNS According to temporal neighborhood suppression, the relevance score of a sample should not be influenced by samples from its temporal neighborhood. For the discriminative setup, this means that samples from $\mathcal{N}^T(x_i)$ should not be trained on when scoring x_i. This can be achieved by simply replacing the random split into folds (Step 2 in Algorithm 1) with one that assigns temporally related content to the *same* fold.

4.5 Experiments using Relevance Filtering

In the last section, relevance filtering has been proposed as a strategy to overcome label noise in concept detection training sets, and extensions of two standard techniques (kernel densities and SVMs) have been presented. In practice, however, such an automatic filtering — which is entirely based on the distribution of content in feature space — is not 100% accurate. Therefore, it is investigated in the following how well relevant content can be separated from non-relevant one, and whether the performance of concept detection can be improved this way. A similar setup to the one from Section 4.3 is used, i.e. training sets with known, varying noise level are randomly compiled. It is demonstrated that the filtering of non-relevant content is possible (though far from perfect), and performance improvements are achieved compared to an equivalent supervised system.

4.5.1 Controlled Setup

The setup of this experiment is almost identical to the one used in Section 4.3: the same randomly sampled training sets and test sets are used. Also, the same feature representation (visual words, followed by a dimensionality reduction using PLSA) and statistical models (kernel density estimation and Support Vector Machines) are employed. It should be noted that in this setup, false positives are drawn from an overall "world distribution" of non-relevant content (which has been denoted with p^0). This corresponds to the modeling assumption made by relevance filtering. However, we will see later that this is not necessarily true in practice, and will provide experimental results on raw web video data. Besides the control runs used in Section 4.3.2, additional results for relevance filtering extensions are presented:

1. **Ground Truth**: The control run from Section 4.3.2 trained on ground truth labels. The system can be viewed as a perfect relevance filtering, where all non-relevant content is correctly relabeled while all relevant samples are preserved.

2. **Weak Labels**: The run from Section 4.3.2, i.e. standard supervised learning trained on weak labels. This can be interpreted as a special case of relevance filtering, with the relevance prior set to 100%.

3. **Relevance Filtering — Fixpoint Iteration**: The relevance filtering extension of the generative kernel density approach from Section 4.4.2, using training variant 1 (fixpoint iteration). The number of training iterations

is set to 100. The relevance prior is set to the correct fraction of relevant material (the behavior when varying this parameter will be studied later).

4. **Relevance Filtering — MCMC**: The relevance filtering extension of the generative kernel density approach from Section 4.4.2, using training variant 2 (MCMC sampling). The cooling schedule of training is set to $\delta = 10^{-5}$. The parameter M (which regulates the influence of the beta prior) is set to half the training set size. The relevance prior α is set to the correct fraction of relevant material.

5. **Relevance Filtering — SVM Self-training**: The relevance filtering extension of the discriminative approach from Section 4.4.3 using SVMs as base classifiers. 10 false positives are filtered in each training iteration. The same smoothness parameter $\sigma = 25$ and cost parameter $C = 5$ are used as in Section 4.3. The relevance prior α is set to the correct fraction of relevant material.

We first visualize the effects of relevance filtering and illustrate what content is identified as non-relevant by the system. Training content is ranked by its score β_i, and the images with highest scores and lowest scores are displayed in Figure 4.7 (a training set with $\alpha = 0.3$ was used, and relevance filtering was run with fixpoint iteration optimization, a bandwidth of 0.275, and a relevance prior of 0.3). For each concept, we see the content that the system identifies to be most relevant (i.e. with the highest scores β) at the top. Below this, material is illustrated that was labeled with the concept but was identified to be non-relevant. Obviously, the content identified as relevant is in fact very likely to be visually related to the concept, and non-relevant material — though labeled with the target concept — tends to be identified successfully. Yet, the quality of filtering is far from perfect, and is also related to the difficulty of the concept: for example, the filtering results for "swimming" are far more accurate than the ones for the challenging concept "cats".

Quantitative results are illustrated in Figures 4.8(a) and 4.8(b) (for the generative model) and in Figure 4.8(c) (for the discriminative one). With decreasing α — i.e. with increasing label noise in the training set — the quality of all detectors (except for the one trained on ground truth labels) degrades significantly, as was already observed for the supervised setup in Figure 4.3. We now examine how relevance filtering performs for the generative kernel density model (Figures 4.8(a) and 4.8(b)). When comparing the two training strategies — fixpoint iteration and MCMC sampling — the fixpoint iteration scheme performs clearly better:

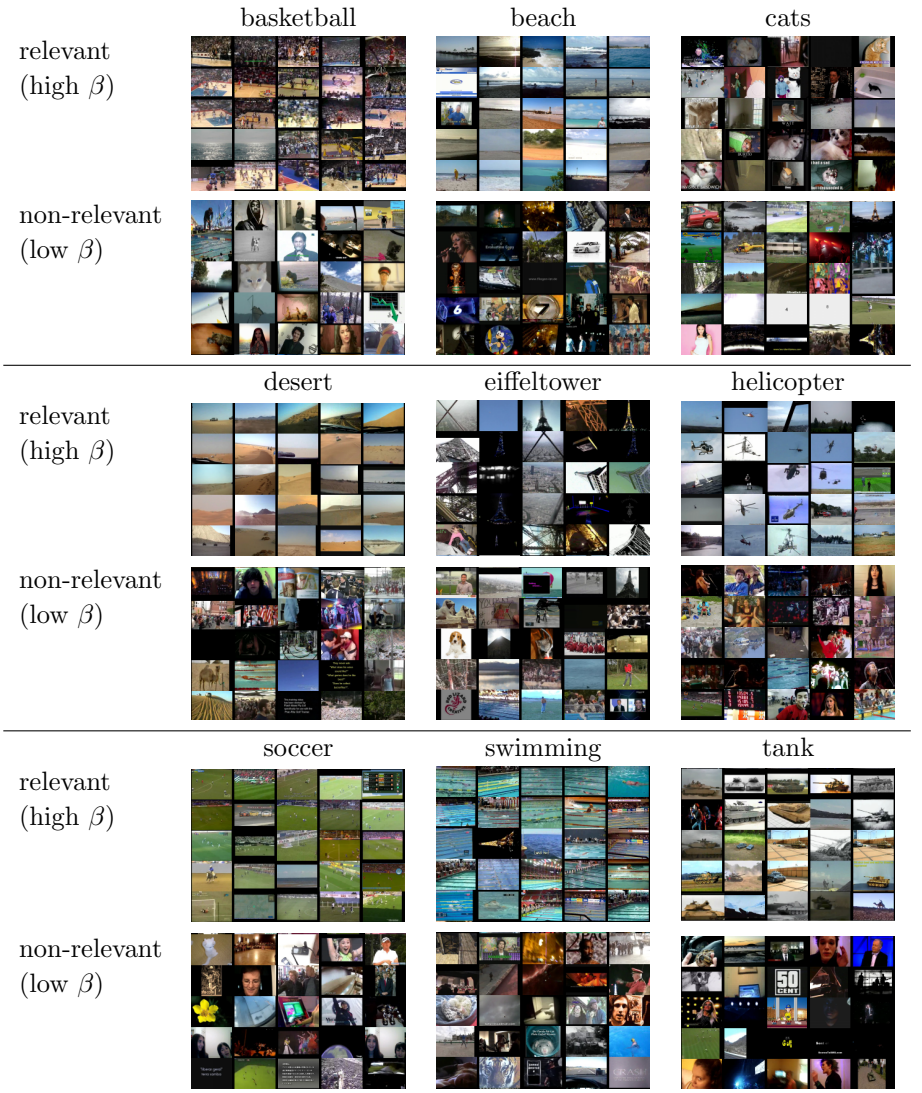
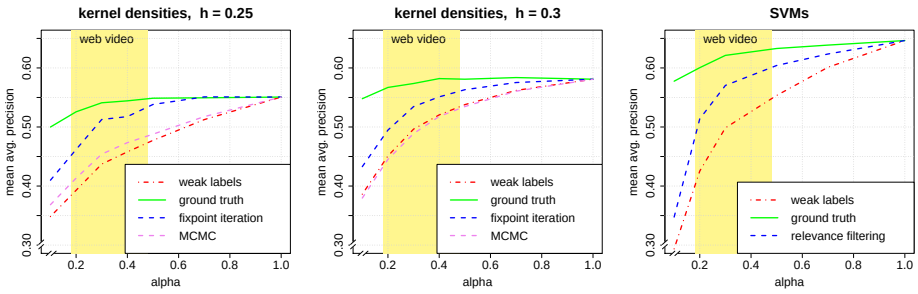


Figure 4.7: Results of relevance filtering: for nine concepts, the frames are displayed that the approach learns to be most relevant (top) and least relevant (bottom). Relevance filtering works in general, though the quality of filtering is strongly related to concept difficulty, as can be observed when comparing "cats" (top right) with "swimming" (bottom center). Pictures from YouTube.



(a) (b) (c)

Figure 4.8: Results of relevance filtering for kernel densities (a,b) and SVMs (c). The performance is plotted against the relevance fraction α ($n = 100,000$). It can be seen that relevance filtering — though not achieving the performance of a hypothetical perfect filter — gives significant improvements over its supervised equivalent.

while MCMC gives only minor ($h = 0.25$) or no improvements ($h = 0.3$) over the supervised case, training with fixpoint iteration improves results significantly. For example, for a bandwidth $h = 0.25$ and a relevance fraction of $\alpha = 0.3$, relevance filtering leads to an improvement from 44% to 51%. When comparing the two kernel bandwidths, we see that relevance filtering gives stronger improvements for the lower bandwidth of 0.25. This can be explained by the fact that the supervised baseline is more competitive due to a stronger smoothing.

For the discriminative SVM approach, similar observations can be made as for kernel densities: relevance filtering does not reach the performance of the oracle-based control run on ground truth labels. Yet, it outperforms standard supervised learning approaches distinctly. Overall, the improvements by relevance filtering in the range of $\alpha \in [0.2, 0.5]$ are significant for all methods (sign test over rank improvement, level 99%).

Finally, the experiment also indicates for which noise levels relevance filtering is the most promising. On the one hand, if the training set is already accurately labeled ($\alpha \approx 1$), standard supervised learning performs quite well, and only minor improvements by relevance filtering are observed. On the other hand, if the training set is extremely noisy ($\alpha \leq 10\%$), relevance filtering becomes difficult. This can be observed in Figure 4.8(c), where for the leftmost point ($\alpha = 10\%$) the improvement of relevance filtering is only weak. Obviously, for moderate values of $0.2 \leq \alpha \leq 0.5$, the benefits of relevance filtering are most prominent, which

Figure 4.9: Non-relevant content from web videos labeled with "Eiffel Tower" does not show the concept, but is obviously correlated it. This renders a fully automatic relevance filtering based only on the frequency in feature space a difficult challenge (pictures from YouTube).

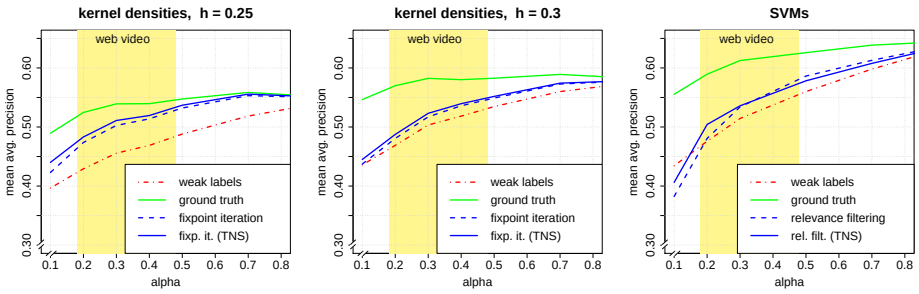
makes relevance filtering particularly interesting for web video. Here, relative performance improvements of up to 17.3% are achieved.

4.5.2 Concept-related Noise Content

In the last experiment, we have seen that relevance filtering can successfully identify non-relevant content, filter it, and thus improve concept learning under label noise. Yet, the experimental setup was restricted in two ways. First, the fraction of relevant content, though not known in practice, was assumed to be given. A simple workaround for this is to set the relevance prior to a "reasonable" value like 0.5, which will be demonstrated to give comparable results to using the true relevance prior.

The second issue is related to the non-relevant samples themselves. While the proposed approach assumes such false positives to be drawn from an overall "world" distribution, non-relevant content depends strongly on the concept in practice. For example, non-relevant material in "basketball" videos tends to show scenes of a cheering crowd, while non-relevant "eiffeltower" material includes urban scenes of Paris (samples "Eiffel Tower" are displayed in Figure 4.9). Note that — as relevance filtering is entirely based on the fact that relevant content forms clusters in feature space — non-relevant material forming similar clusters (like "shots of Paris") may be difficult to separate from truly relevant content. Therefore, this experiment will use concept-related noise content as it occurs in real-world web videos.

Setup We use a similar setup as in previous experiments, i.e. training sets of the same size are randomly compiled for different noise levels. The key difference is that false positives — which were previously sampled from videos *not* labeled with the target concept — are now drawn from clips tagged with the concept, but were



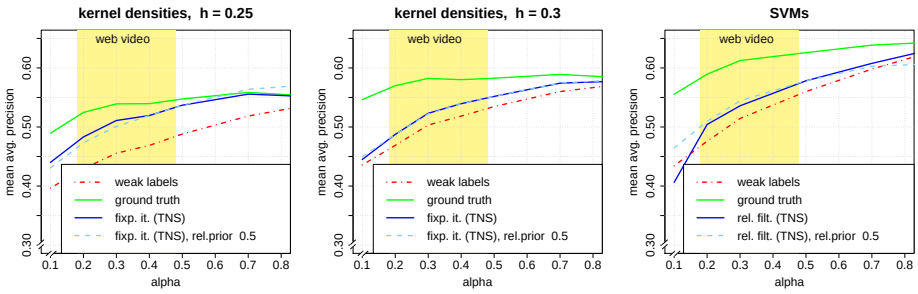
(a) (b) (c)

Figure 4.10: Results on raw web video content, where non-relevant material is correlated with the target concept ($n = 100,000$). Relevance filtering still outperforms supervised learning, though by a lower margin.

assessed manually to be non-relevant. Similarly, the test set was adapted, i.e. 500 concept-related non-relevant frames were included. Again, the same feature representation (visual words and PLSA) and statistical models (kernel densities and SVMs) were tested. Fixpoint iteration — which was found to clearly outperform MCMC previously — was used to train generative relevance filtering.

Results Like in previous sections, results are reported in terms of mean average precision plotted against the relevance fraction α (Figures 4.10 and 4.11). The first — and most important — observation is that relevance filtering still outperforms supervised learning significantly (sign test, level 99%), but by a lower margin compared to previous experiments. While relative improvements by relevance filtering reached up to 17.3% (compare Figure 4.8), they are now below 11.8%, and for SVMs an aggressive filtering at high noise ratios ($\alpha = 0.1$) even gives a lower performance than supervised learning (Figure 4.10(c)). This is because false positives, which are now correlated with the target concept, are more difficult to identify and filter.

Figure 4.10 also illustrates the effect of temporal neighborhood suppression (TNS). It can be seen that using this video-specific approach to suppress evidence from temporally close content, improvements in the range of 1% can be achieved for the generative case compared to a plain image-based system. This improvement is minor but statistically significant (sign test, level 99%).



(a) (b) (c)

Figure 4.11: Comparing relevance filtering on raw web video training sets when using the correct relevance prior (dark blue) with using $\alpha = 50\%$ (light blue). It can be seen that the latter choice leads to comparable results.

Finally, we address the question how to estimate the relevance prior α. A simple solution is proposed, namely to set it to a "reasonable" value of 0.5 (which is typical for web-based training sets as shown in Section 4.3). Figure 4.11 compares relevance filtering (with temporal neighborhood suppression switched on) when using the true prior and when using $\alpha = 0.5$. It can be seen that setting $\alpha = 0.5$ (i.e. by filtering half of all positive training samples) gives a performance comparable to the true value, at least for the range of $\alpha = 0.2 - 0.5$ that is typical for web video. For SVMs and very noisy training sets, this choice even outperforms an aggressive filtering.

4.6 Discussion

In this chapter, *relevance filtering* has been presented, a novel approach for concept learning from user-tagged video. This method addresses the fact that web video tags are coarse, subjective, and context-dependent, such that significant amounts of training material do not show the target concept to be learned. This degrades the performance of standard concept detectors.

To address this problem, relevance filtering views web tags as weak indicators of true, latent concept labels and infers these during training. This approach can be used as a wrapper around generative base models (as demonstrated for kernel densities) as well as discriminative ones (as shown for SVMs).

We have tested relevance filtering in two setups. In the first one, non-relevant content was – according to the modeling assumptions – drawn from an overall world distribution. Here, relevance filtering systems were able to identify non-relevant content reliably and give strong improvements over their supervised counterparts. In the second setup, the system was trained on raw web video content, where non-relevant material is often correlated with the target concept (for example, noise material tagged with "Eiffel Tower" tends to show urban scenes of Paris). Here, relevance filtering still gave improvements, though by a lower margin. This can be explained by the fact that correlated noise content forms clusters in feature space similar to relevant material, and is thus difficult to identify.

On the one hand, these results indicate that the problem of label noise in web data can be overcome to some extent, as the proposed relevance filtering gives improvements over standard supervised learning. On the other hand, systems trained on ground truth labels still perform significantly better. This indicates that — if we were able to do a better filtering — we could improve concept detection on weakly labeled training data even further. Correspondingly, a promising direction of future work is to integrate the current (fully automatic) approach with moderate manual supervision. This way, a better filtering is to be expected: more reliable label information can be achieved this way, as well as better estimates for the relevance fraction α. Particularly, active learning techniques — where the system selects informative examples for the user to label [AQ08, Set09] — might be investigated. Such an extension fits quite elegantly into the proposed framework: whenever a user annotates a training sample x_i, the relevance score β_i is adapted accordingly, and learning is re-iterated. Also, relevance scores β_i can directly be used as a criterion for selecting query samples for manual labeling, using uncertainty sampling [LG94] or related strategies. First experiments conducted by the author of this thesis already demonstrate significant improvements using such *active relevance filtering*.

Another interesting extension to the current approach might be to make better use of the temporal structure of video. This is currently done by a suppression of the temporal neighborhood, i.e. content appearing at about the same time / in the same clip is ignored when judging relevance. On the other hand, this content also tends to be similarly relevant, such that scores might be enforced to be smooth over time. This is ignored by the current approach, and an extension that uses temporal correlation might be an interesting direction to improve relevance filtering.

Chapter 5

Style Modeling for Concept Detection

In this chapter, the proposed framework for concept learning from user-tagged web content is extended further, based on the fact that users at portals like Flickr and YouTube organize material in categories. This information is used to integrate concept detection with *style modeling*: a distinct annotation model is learned per category (or *style*, respectively). To use these models for annotation, test images are assumed to come in style-coherent groups, and pictures are mapped to a style using their *context* (i.e. other images from the same batch).

The key contributions of this chapter are[1]:

1. A novel algorithm for image and video annotation is proposed which makes use of category information in web portals, and is the first one that combines autoannotation with style modeling.

2. It is shown that context information — if available and used as proposed — improves concept detection significantly. In experiments on the COREL dataset and Flickr photos, relative performance improvements of up to 100% are reached compared to an annotation of individual images ($n = 32,000$).

3. On the COREL-5K standard benchmark, the approach achieves a competitive performance (mean per word precision/recall: 25% / 39%).

[1]This chapter is based on the author's work in [DUBW09]

5.1 Introduction

Concept detection (or autoannotation, respectively) is concerned with automatically detecting the presence of objects, scene types, activities, etc in images and videos. The task poses an extraordinarily difficult challenge to computer vision systems, because hundreds or thousands of concepts are to be distinguished, and because intra-class variation is enormous — for example, views of a concept like "dog" may vary with changes of object pose, camera perspective, illumination, and finally with inherent variation of object instances themselves. Yet, though the performance of concept detectors remains far from human accuracy, the approach is attributed a high potential for image and video retrieval [SWdR+08] and has been realized in a variety of prototypes [CHL+07, LW08, SvZ08, Sno07, WZLM08].

A key problem with autoannotation is that the effort associated with training data acquisition is high, as the number of classes to be learned may be in the range of thousands. This opens the question whether novel sources of information that are freely available can be employed to improve the performance of detectors or to reduce the manual annotation effort associated with training. One such information source is the web, which offers vast databases of visual content enriched with user-generated textual descriptions. For the domain of images, Flickr[2] and text-based image search engines have been pointed out as interesting information sources, and a variety of approaches [FFFPZ05, LWFF07, WZLM08] have been suggested to employ them for visual learning. For the video domain, portals like YouTube offer similar possibilities, and it has been demonstrated in Chapter 3 that web video offers a scalable, flexible, and efficient way of concept learning.

While the aforementioned methods focus on web-based data as additional training *content*, portals such as YouTube or Flickr also offer information of a *structural* kind. This is because users do not only enrich their images and videos with descriptive tags, but also assign them to semantic classes: content at YouTube is organized in 15 categories such as "Travel&Events" or "Sports". Similarly, images at Flickr are assigned to thousands of *groups* dealing with topics like "New York City" or "Macro Photography" [NGP08]. This is not limited to web content: as the desire to categorize and organize is deeply human, users also create folders on their local harddrives and place pictures in them that "belong together". Like Flickr groups, these batches of images may be associated with certain events or locations, being captured over the latest holiday trip or containing photos of a new-born baby. Similarly, personal video content is compiled to clips, which may show a friend's wedding or a soccer game. Even more generally, video digest of

[2] www.flickr.com

(a) (b)

Figure 5.1: Context information improves image and video annotation: in both cases, an automatic tagging is difficult when only looking at an individual item. For an image showing an outdoor scene (a), potential tags might be "forest" or "park". A head shot scene in a video (b) might be labeled "monologue" or "interview". By taking context into account, these conflicts can be disambiguated: since other pictures in the same group show mostly urban scenes, the correct tag "park" is inferred. The video scene in question appears in a news context, such that the preferable tag turns out to be "interview". Pictures from Flickr/YouTube/ZDF.

all kinds comes in temporal units. On a low level, these are frames and shots, but there are also units on a higher level, like scenes, shows, clips, and films. What we learn from these examples is that — in a variety of practical situations — visual content comes in groups and categories.

The key hypothesis of this chapter is that this information can be used for an improved concept detection. Thereby, other images from the same group serve as *context* for an item to be annotated. This idea is illustrated in Figure 5.1 for both photos and video: for the image case, a group of pictures is shown from a weekend trip to Rome. Consider the image at the bottom right showing an outdoor scene with trees and greenery. Using evidence from this single image only, a concept detection system might confuse the tags "forest" and "park". However, if further taking into account that the image belongs to a group showing mostly urban scenes, this ambiguity can be resolved, and the correct tag "park" can be inferred. The same holds for the video content displayed: a frame is highlighted which — if viewed individually — might be labeled both as an "interview" or a "monologue". If viewed in a "news" context, however, the preferred tag turns out to be "interview". Both examples indicate that context information might help to improve concept detection.

Figure 5.2: Illustrating parallels between image annotation (left) and handwriting recognition (right): in both cases local ambiguity may occur (boxes in the center), i.e. individual samples are difficult to classify (on the left an image might be labeled with “forest” or “park”, on the right a digit might be a “1” or a “7”). This ambiguity can be resolved using context information: for the picture in question, the fact that other images in the batch show mostly urban scenes helps to infer the tag “park”. In the case of handwriting, alternative versions of 1s and 7s are found (pictures from Flickr/MNIST handwritten digits).

Unfortunately, context information has been widely neglected by image and video annotation so far: most methods assume images and keyframes to be independent and identically distributed (which is incorrect) and label them one by one using the same statistical model [CCMV07, DBdFF02, FML04, LMJ04, LW08, WZLM08]. This approach is limited in the sense that the context of an item to be annotated is not taken into account at all.

To overcome this restriction, the goal of this chapter is to improve autoannotation by employing additional context information. Therefore, we turn towards solutions developed in other domains, namely handwriting recognition and optical character recognition (OCR). There are strong parallels between concept detection and character recognition: similar to pictures or frames (which have been pointed out above to come in correlated batches), written glyphs form coherent groups, like sentences, pages, or documents. Correspondingly, context plays a similar role in both domains, as is illustrated in Figure 5.2: in the box on the left side, the ambiguous image from Figure 5.1 is displayed again, and it can be seen that different tags would be appropriate if viewed in a “City” context (namely, “park”) or in a “Nature” context (namely, “forest”). On the right, a “difficult” handprinted

digit is displayed (a decision based only on the digit itself is highly ambiguous and might be either "1" or "7"). Taking into account context information drawn from the complete document, this ambiguity can be resolved. Different versions of "1"s and "7"s are found, such that obviously the upper row corresponds to an American-style writer (where the letter in question is to be interpreted as a "7"), while the lower one has been written in European style (and the letter is a "1"). In both domains, context resolves local ambiguity.

In OCR, context has successfully been exploited by *style modeling* approaches [MB02, SN05]. These methods assume the glyphs within a document to be drawn from a common latent source (or *style*). A style is defined as a category of samples sharing a coherent appearance — in case of OCR, this corresponds to all glyphs written by a certain person or printed in a certain font. Given this style information, classification is performed in three steps. First, for each style a specific annotation model learned, using training samples drawn from the style only. Second, to perform recognition of a test document, context information is used to reliably infer the style of the whole document. Third, for each character the same style-specific classifier is used for a highly accurate recognition. Using this approach, it has been demonstrated for handwriting that style information can improve recognition significantly [MB02, SN05].

It seems reasonable that the same might be possible for the domain of concept detection. Therefore, this chapter adopts style modeling for image and video annotation. Similar to the OCR domain, different styles will be defined as categories of samples (here, images) sharing a distinctive visual appearance. This common appearance may be due to various reasons — we can use styles associated with certain holiday trips (like "Sightseeing" or "Safari"), objects of interest (like "Portrait" or "Landscape Photography"), or photographic or dramatic categories (like "Macro Photography" or "Action Movies"). All these examples define categories of images/frames sharing a coherent appearance.

It should be noted that this definition of styles as generic image categories is different from previous understandings in the multimedia literature, which refer to style as the product of an authoring process (for more information, please refer to Snoek et al.'s work [SWG+06]). It should also be kept in mind that styles are *not* identical to annotations. Though there may be strong relations between both (for example, the tag "portrait" might be a good choice for pictures from a "Portrait" style), this does not necessarily have to be the case. Images in a style may be associated with very different annotations — for example, a "Safari" style might show close-ups of animals as well as panoramic views.

Let us now address the practical realization of style modeling for concept detection. One particular problem is that a learning of annotation models for different styles requires style-specific training images, i.e. the overall training set must come with additional "style labels". To acquire this information, we turn towards web content again. Here, portals like YouTube or Flickr do not only offer tags and descriptions (which we employ to train annotation models), but pictures and videos are also organized in semantic categories (which we will use to infer style labels). Particularly, Flickr offers thousands of *groups*, which cover a rich space of specific semantics [NGP08], ranging from *geographical/event groups* ("Switzerland", "Live Music") over *content groups* ("Leaves", "Cats") to *visual style groups* ("Life in Black and White", "Macro Photography"). We will use such groups as equivalents to styles, which allows us to acquire style-specific training data by simply downloading it.

This constitutes a novel approach towards web-based autoannotation, which achieves an improved annotation accuracy using style modeling. The approach comprises of three steps:

- **Learning**: A number of style-specific image annotation models is learned from web categories. For example, for tagging users' holiday snapshots, styles like "Sailing Trip" or "Safari" are learned.
- **Style Inference**: Given a group of previously unseen images, a style decision is made based on the whole batch. For example, the system infers that pictures match the "Sightseeing" style.
- **Style-specific Annotation**: The style-specific model (here, "Sightseeing") is applied to each image within the group, obtaining an accurate annotation.

As already mentioned, Flickr provides an excellent data source for applying this framework, as it offers rich group annotations for learning a wide range of specific styles. Therefore, the remainder of this chapter will focus on the domain of still images, with style information provided by Flickr. Similarly, notation is adopted for the rest of this chapter: pictures and video frames will both be referred to as "content", "samples", or — for the sake of simplicity — "images".

The proposed framework will be evaluated on the COREL dataset and real-world photo stock downloaded from Flickr. In these experiments, styles correspond to different travel destinations, i.e. an annotation of personal holiday snapshots is simulated. Significant performance improvements will be reported by style modeling. These are demonstrably achieved by using context information, which has been neglected by most methods so far.

This chapter is organized as follows: first, an overview of related work on style modeling and on automatic image annotation is given in Section 5.2. Next, the proposed approach is presented (Section 5.3), and experimental results are provided in Section 5.4. Section 5.5 concludes the chapter with a discussion and outlook.

5.2 Related Work

Since the proposed style modeling approach is adopted from the domain of handwriting recognition, this section starts with a discussion of related work in this area. After this, an overview of image annotation will be given, and particularly approaches related to style and context information will be addressed.

5.2.1 Style Modeling

Style modeling has previously been used for an improved character recognition based on the fact that letters from the same document usually share a coherent font and degradation. This is referred to as the *style* of a document. To make use of this information, Baird and Nagy [BN94] have presented an approach that adjusts a pre-defined multi-font base classifier in a self-training: iteratively, the system is applied to the test document, and classification results are fed back as labeled samples to system training. This way, the classifier is adapted to the target document.

The *style consistency* model proposed by Sarkar and Nagy [SN05] assumes a limited number of discrete styles to be given. For each style, a specific classifier is trained, and test documents are mapped to a style using a maximum-likelihood approach over the whole batch. Finally, the style-specific classifier is used for an accurate annotation.

While this approach has been demonstrated to give a superior performance compared to omni-font classifiers [SN05], a problem remains in fonts that have not been seen in training. In this situation, style consistency can at best be expected to map a test document to the "most similar" style learned. A more general approach based on *hierarchical Bayesian* methods [MB02] has been proposed by Mathis and Breuel. Here, style is not a discrete variable, but becomes a parameter of the sample generation process. This parameter is itself drawn from a hyperprior modeling the distribution of styles. For each test document, a common style parameter is sampled, which again guides the character generation process. An improved generalization to new fonts for an OCR scenario has been validated [MB02].

Finally, cluster-and-label approaches have been proposed for style adaptation. As the test document shows a coherent style, it is assumed that a clustering of its characters can be applied, such that cluster boundaries coincide with decision boundaries. Breuel proposes two techniques for clustering: the first one is based on an EM estimation of Gaussian Mixture Models [Bre01b]. The second approach uses an additional learning step to infer an improved similarity measure from labeled training data: a Multilayer Perceptron (MLP) is used to model $P(c_1 = c_2|x_1, x_2)$, i.e. the posterior for two arbitrary samples x_1, x_2 to show the same character. These scores are computed over all pairs of test samples, and are used to compute the final posterior $P(c|x)$ using a simulated annealing procedure.

5.2.2 Image Annotation

Starting with Mori et al's pioneering work on automatic image annotation [MTO99], a variety of approaches has been suggested. Since potential annotations (or *tags*) are often associated with certain regions, images are correspondingly viewed as collections of local parts $\mathcal{V} = \{v_1, ..., v_n\}$. These can be obtained using an image segmentation approach [LLM03] or by a sampling of local patches [FML04]. The goal of image annotation is to map this description $\mathcal{V}$ to tags t from a predefined vocabulary T. Since tags of interest can include all kinds of semantic concepts, image annotation unifies related tasks such as object category recognition [EVGW+07] and scene recognition [QMO+07].

Three general approaches towards image annotation can be distinguished. Models of the first category — referred to as *joint probability models* in the following — are targeted at estimating a joint distribution $P(v, t)$ of local image features v and annotations t. Based on this idea, Mori et al. [MTO99] suggested to discretize patches into *visual words*, such that $P(v, t)$ turns out to be a probability table. Recently, a similar approach has proven successful that learns a large-scale visual vocabulary in the order of millions of visual words, which is made feasible by a hierarchical extension of K-Means clustering [FM08].

To overcome drawbacks in terms of sparseness for limited training data, it has been proposed to model the joint distribution $P(v, t)$ using *topic models* [Hof01]. These assume tags and patches to be sampled independently from latent semantic aspects (or *topics*) z:

$$P(v, t) = \sum_z P(z) \cdot P(t|z) \cdot P(v|z)$$

Topics can be inferred from a collection of labeled training images using Probabilistic Latent Semantic Analysis (PLSA) [MGP04], Latent Dirichlet Allocation

(LDA) [FFP05], or a variety of hierarchical models [BDF+03]. Alternatively, one aspect z can be associated with each single image, which leads to a series of *relevance models* suggested by R. Manmatha and co-workers [FML04, LMJ04, LLM03]. These models differ in the realization of patch distributions $P(v|z)$ (which can be continuous [FML04] or discrete [LLM03]) and word distributions $P(t|z)$ (which can be multinomial [LLM03] or multiple Bernoulli distributions [FML04]). Another alternative is to model class-conditional densities $p(v|t)$ for local regions associated with a tag. Carneiro et al. [CCMV07] proposed Gaussian mixtures for this purpose, which are estimated in an efficient manner by first computing image-wise mixtures and then fusing them in a clustering.

The second category of *global methods* uses global image-level features for a tag decision. One simple, popular strategy based on such global descriptors is — given an image to be labeled — to find similar training pictures and adopt their tags. Several prototypes have been based on this idea [LW08, LCZ+06, RvBKB08, WZLM08] which vary in the concrete features and distance measures used. Other methods using global descriptors are based on discriminative strategies like Maximum Entropy [JM04], SVMs [MCSM07], or kernel densities [YSR05].

Finally, approaches from the third category view image annotation as a *weakly supervised learning* problem: it is assumed that the presence of a tag is caused by a certain region in the image, and concept detection involves the explicit identification of this "relevant" region (such that tags can be assigned directly to regions instead of images). Duygulu et al. [DBdFF02] adopt a machine translation setup for this purpose: visual features and tags are interpreted as different representations of an image, similar to roughly aligned texts in different languages. Exact correspondences between image regions and tags are inferred using Expectation Maximization (EM). Following a similar idea, *multiple instance learning* has been applied to image tagging. Thereby, pictures are viewed as *bags* of image regions. Tag-related regions are identified using Diverse Density [MR98] or adaptations of supervised support vector machines [YDF05]. In a similar fashion, Kück et al. [KCdF04] cast the assumption that at least one relevant region can be found in each labeled image, which leads to a constrained semi-supervised learning problem. A probabilistic framework is suggested, which is solved using Markov Chain Monte Carlo (MCMC) sampling and provides annotations on region level.

While all these methods differ in terms of features and underlying statistical models, they share one important drawback, namely that images are treated individually. In contrast to this, the focus of this chapter is on the use of context information. The proposed approach can be used as a *wrapper* around existing probabilistic image annotation models and can thus be integrated with several

methods mentioned above (for example, with the ones from the first category called *joint probability models*).

5.2.3 Autoannotation using Context and Style

As outlined above, the majority of current image annotation approaches treats images and video frames individually. Yet, there is some work — both for images and video — that exploits information beyond single images and thus makes use of context or style.

For the video domain, Snoek et al.'s *Semantic Pathfinder* [SWG+06] includes additional levels of analysis based on *style* and *genre* information. The key idea is that a video is created in an authoring process, and that correspondingly the appearance of semantic concepts is accompanied by a certain cinematographic style. A variety of features is used to exploit this fact (like shot length, the detection of closeups, etc) and is demonstrated to give improvements over a content-only baseline system. We follow the idea that style information can help to improve concept detection. However, in contrast to the Pathfinder framework, the notion of style in this chapter is different: while style in [SWG+06] refers specifically to characteristics of the authoring process (and corresponding features are used), in this chapter styles correspond to generic categories of content, and it is abstracted from why content shares a certain style or what characterizes a style (this information is derived in a learning step).

For the domain of images, a variety of methods has been proposed using the structure of content beyond individual items. Li et al. [LSW08] address the image annotation problem using photo-sharing websites such as Flickr. They address the fact that tag information is noisy and filter non-relevant tags. Therefore, the user structure at Flickr is taken into account in the sense that tag relevance is related to the number of different users assigning a tag. In a similar approach, Mei et al. [MWH+08] exploit the tag structure of image datasets to learn a so-called "semantic similarity" based on tag correlations. These approaches are similar to the work presented in this chapter in the sense that the structure of an image collection is taken into account. In fact, Flickr groups will be used as a similar information source in this chapter. The key difference, however, is that — while these approaches are targeted at making a better use of *training* data (by filtering annotations or learning better similarity measures), the work presented in this chapter focuses on the use of context during *testing*. This makes the proposed approach orthogonal to training set improvements as in [LSW08, MWH+08], and ultimately concept detection could be given a boost by combining both approaches.

There are other approaches more similar to this work in the sense that content structure is taken into account in testing. For the video domain, context plays a key role for the detection of *events*, which are typically composed of several subitems occurring in a temporal surrounding or in a certain order. Snoek and Worring [SW05a] propose to make use of such context in soccer and news video (events are "goal", "interview", etc). Multi-modal detectors for certain clues mine the temporal surrounding of a potential event E (like "a closeup appears shortly after E"), and these binary detections form the input to a statistical classifier. While this chapter follows this idea of using context to disambiguate recognition, the proposed framework differs in two aspects: first, while in [SW05a] a full spectrum of temporal relationships is modeled, only a grouping of content is assumed to be given. Second, our framework addresses the detection of generic concepts and does not focus on events in video.

Other approaches for the domain of still images make similar use of context in the test data [GNC+08, NYGMP05]. Gallagher et al. [GNC+08] match pictures with events in personal calendars (like "George's Wedding"). Naaman et al. [NYGMP05] group photos to events at photo sharing sites and use this information for person identification. Cristani et al. [CPCM08] enhance image annotation with a latent variable that models the geographic region in which a picture was taken. A collection of geo-tagged pictures is then clustered into regions of geographic proximity, and it is demonstrated that geo-localization can be solved better when using a whole batch of multiple pictures taken at the same location instead of a single one.

What we learn from these approaches is that a grouping of pictures can be inferred from meta-data such as times and locations of capture. Our style model relies on the very same grouping information. Further, these contributions show that for specific recognition applications it can be helpful to view images in groups instead of individually. However, it should be pointed out that this chapter addresses the more general annotation problem for generic concepts. The notion of styles here is a generic one — it can refer to geographic categories as well as to semantic or technical ones, and is learned automatically from Flickr groups.

Finally, another closely related approach is the one by Cao et al. [CLKH08], who propose a hierarchical annotation model in which pictures are clustered to *events* previous to annotation. The system groups test content based on time and GPS stamps. Similar to this chapter, Cao et al. also address the annotation of images, and their events resemble the notion of *style* used here. Yet, the approach differs from the work in this chapter with respect to several aspects: Cao et al. emphasize the challenge of how to obtain grouping information of pictures using time and GPS

stamps. Context is used in form of correlation terms in a postprocessing step. Instead, the approach proposed here assumes a grouping of images to be given and focuses on how this information can improve autoannotation. Therefore, image annotation itself is directly adapted, and it is demonstrated that style modeling is a well-founded and successful strategy to do so.

5.3 Approach

In this section, a style model from the domain of optical character recognition [SN05] is adopted for image annotation. The key idea is that images should not be labeled individually, but *context* should be employed in form of other images in the same group. Therefore, two assumptions are made:

- Images are assumed to belong to a predefined set of categories (or *styles*), and style-specific training sets are assumed to be available (it was pointed out previously that web-based portals like Flickr can be employed for this purpose).

- Test images to be annotated are assumed to come in groups (or *batches*). This information may be available in form of coherent times of capture or upload, in form of GPS stamps, or may be provided by the user himself (for example, by placing pictures in the same folder).

The framework models style as a latent random variable, and comprises of three key steps: (1) *training*, in which a number of style-specific annotation models is learned. (2) *style decision*, in which the style variable is inferred using context information. (3) *annotation*, in which a style-specific model is used for a precise annotation. In this framework, style modeling serves as a wrapper around image annotation, and can be integrated with a variety of probabilistic image annotation methods (though for this chapter the one by Monay and Gatica-Perez [MGP04] based on Probabilistic Latent Semantic analysis (PLSA) [Hof01] is chosen).

This section is organized as follows: first, basic concepts are introduced, and the general structure of the proposed approach is outlined (Section 5.3.1). After this, a first model based on this structure will be discussed, where style is neglected and the approach boils down to a conventional image-wise annotation [MGP04] (Section 5.3.2). This model will serve as a baseline in later experiments. After this, two realizations including style modeling will be presented (Sections 5.3.3 and 5.3.4).

5.3.1 Basic Concepts

The core of the proposed approach is an image annotation model based on Probabilistic Latent Semantic Analysis (PLSA) [Hof01], a *topic model* that was originally developed in the text domain and has successfully been adopted for a variety of visual recognition tasks [QMO+07, SREZ05, SMH04]. PLSA views documents (or images, in our scenario) as collections of words (here, *visual words*, see Section 3.5). It is assumed that each document is a mixture of a few latent aspects (or *topics*), and that the words of the document are sampled from these topics.

We consider a set D of images to be given, whereas each image d is represented by a set of visual words v from a vocabulary V and by tags $t \in T$. The latent topics are denoted with $z \in Z$. Finally, images are assumed to come in groups sharing a coherent *style* $s \in S$. According to our style-based PLSA model, the generative process of sampling a batch of images D is the following:

1. pick a style s
2. for $d \in D$:

   for $i = 1, ..., n_T(d)$: *(sampling tags)*
   sample $z_i \sim P(z|d)$
   sample $t_i \sim P(t|z_i, s)$
   for $j = 1, ..., n_V(d)$: *(sampling visual words)*
   sample $z_j \sim P(z|d)$
   sample $v_j \sim P(v|z_j, s)$.

$n_T(d)$ and $n_V(d)$ denote the number of tags and visual words for image d. The number of topics $|Z|$ is assumed known and fixed (usually, $|Z|$ is much smaller than the number of documents and words, such that z serves as a "bottleneck variable"). $P(z|d)$ assigns topics to images. This sampling process posits that tags and visual words are conditionally independent given topics z and style s, and that they are drawn from the following distributions:

$$\begin{aligned} P(t|d,s) &= \sum_{z \in Z} P(t|z,s) \cdot P(z|d) \\ P(v|d,s) &= \sum_{z \in Z} P(v|z,s) \cdot P(z|d) \end{aligned} \tag{5.1}$$

Note that these distributions depend on the style s, i.e. both tags and visual words are influenced by the style that was chosen for the whole batch at the beginning

of the sampling process. As this style is the same for all pictures in the batch D, pictures are correlated.

5.3.2 Baseline: Coupled PLSA

While the last section introduced the fundamental structure of the proposed model — with tags and visual words sampled from style-dependent distributions — we now discuss several concrete realizations of this framework making different use of style information.

The first version ignores style information completely: the topic distributions $P(t|z,s)$ and $P(v|z,s)$ are replaced with simpler equivalents $P(t|z)$ and $P(v|z)$, such that the style variable s does not have any influence and can be dropped:

$$\begin{aligned} P(v|d) &= \sum_{z \in Z} P(v|z) \cdot P(z|d) \\ P(t|d) &= \sum_{z \in Z} P(t|z) \cdot P(z|d) \end{aligned} \tag{5.2}$$

This model corresponds to the *PLSA-words* image annotation method proposed by Monay and Gatica-Perez [MGP04]. Like most standard approaches, it treats each image individually. A graphical illustration of the sampling process can be found in Figure 5.3(a).

In the following, learning and inference with this model are briefly outlined, which are both based on a maximization of the overall data likelihood (using the terms from Equation (5.2)), where $n(.,d)$ denotes the number of occurrences of a specific tag or visual word in image d:

$$\mathcal{L}(D) = \prod_{d \in D} \left[P(d) \cdot \prod_{t \in T} P(t|d)^{n(t,d)} \cdot \prod_{v \in V} P(v|d)^{n(v,d)} \right] \tag{5.3}$$

Learning The topic posteriors $P(z|d)$ and topic vectors $P(v|z)$, $P(t|z)$ are learned from a set of annotated training images D. For standard PLSA models, such learning is done by maximizing the overall data likelihood (Equation (5.3)), whereas optimization is carried out using Expectation Maximization (EM) [DLR77] or variants [Hof01]. Iteratively, two steps are applied: first, in the *"E"-step*, the posteriors for latent variables (i.e. the topic that each word is sampled from) are estimated. In the subsequent *"M"-step*, the expected log-likelihood of the training data with respect to these posteriors is maximized, resulting in updates for

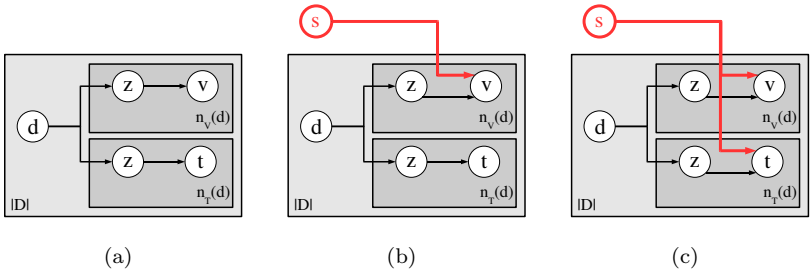


Figure 5.3: Graphical models depicting the sample generation process for image features v and tags t. (a) The baseline model [MGP04]. (b) The "appearance-only" style extension. (c) The "appearance-and-tags" style extension.

the topics and topic distributions. For a detailed description of EM, please refer to Hofmann's work [Hof01]. In this chapter, a slightly altered procedure will be used, for which Monay and Gatica-Perez have reported improved image annotation results [MGP04]:

1. The distribution of visual words is neglected, and the topic distribution $P(z|d)$ is learned by maximizing the likelihood over the training images' *tags* only:

$$\mathcal{L}_T(D) = \prod_{d \in D} \left[P(d) \cdot \prod_{t \in T} P(t|d)^{n(t,d)} \right]. \tag{5.4}$$

   For optimization, the standard EM algorithm is used.

2. PLSA is run over visual words to compute $P(v|z)$. Again, EM is used for optimizing the likelihood (now over visual words only):

$$\mathcal{L}_V(D) = \prod_{d \in D} \left[P(d) \cdot \prod_{v \in V} P(v|d)^{n(v,d)} \right]. \tag{5.5}$$

   Thereby, the topic distributions $P(z|d)$ learned in the previous step are fixed, i.e. they are not changed during the "M"-step of the EM algorithm.

Inference Given a previously unseen batch D^* of test images d^* to be labeled, each image is annotated independently. Thereby, the visual distribution $P(v|d^*)$ is given, and the tag distribution $P(t|d^*)$ is inferred:

1. Given $P(v|d^*)$, $P(z|d^*)$ is computed using a "fold-in heuristic" [Hof01]: the EM algorithm is applied to maximize $\mathcal{L}_V(D^*)$, whereas the topic appearances $P(v|z)$ learned in training are fixed.

2. The distribution of tags is estimated, with $P(t|z)$ learned previously in training:

$$P(t|d^*) = \sum_{z \in Z} P(t|z) \cdot P(z|d^*), \tag{5.6}$$

Finally, a set of tags with highest posterior probability $P(t|d^*)$ is selected as annotations of d^*.

5.3.3 Style Variant 1: Appearance-Only

In this section, a second realization of the style-based image annotation framework from Section 5.3.1 is presented. While the baseline version introduced in the last section labels images individually, the model presented in the following takes the style of image batches into account.

Therefore, the visual word distribution $P(v|z)$ from the baseline model is replaced with style-specific appearance models $P(v|z,s)$. The model for image d turns out to be:

$$\begin{aligned} P(v|d,s) &= \sum_{z \in Z} P(v|z,s) \cdot P(z|d) \\ P(t|d) &= \sum_{z \in Z} P(t|z) \cdot P(z|d) \end{aligned} \tag{5.7}$$

Note that — while the appearance model of visual words is now style-dependent as in Equation (5.1) — the distribution of tags $P(t|d)$ remains as in the baseline model. A graphical representation of the resulting sampling process is illustrated in Figure 5.3(b), where changes relative to the baseline model (Figure 5.3(a)) are highlighted. Two things should be kept in mind: first, only a single style variable is drawn for the whole group, i.e. all images in the batch share the same style. Second, as the tag distribution $P(t|z)$ remains unchanged, it is implicitly assumed that tags appear with the same frequency in all styles but appearance differs between styles (for example, the tag "building" looks different in a "New York City" style and in an "Africa" style). We will refer to this approach as the "appearance-only" style model in the following.

Learning Like for the baseline model, the topic posteriors $P(z|d)$ and topic vectors $P(v|z)$, $P(t|z,s)$ are learned using a two-step procedure similar to the one for the baseline (Section 5.3.2). First, standard EM on the *tags* of all input images (regardless of style) is used to learn $P(t|z)$ and $P(z|d)$ by maximizing the tag likelihood (Equation (5.4)). Second, the distribution of visual words $P(v|z,s)$ is learned. For this purpose, it is assumed that the style $s(d)$ for each training image d is given (it has already been pointed out that Flick groups will be employed for this purpose). For each style s, the following likelihood is maximized:

$$\mathcal{L}_V^s(D) = \prod_{d:s(d)=s} \left(P(d) \cdot \prod_{v \in V} P(v|d,s)^{n(v,d)} \right) \tag{5.8}$$

Again, optimization of the topic appearances $P(v|d,s)$ is carried out using EM, whereas the topic distributions $P(z|d)$ are fixed.

Inference Compared to inference in the baseline model (Section 5.3.2), the key difference is that the style variable s is unknown. As Sarkar and Nagy demonstrated, a globally optimal Bayesian inference is infeasible [SN05]: since tags and style are both unknown and influence each other, optimal inference requires to test all combinations of tags, whose number grows exponentially with the number of test images in a batch.

To solve this problem, a similar strategy is followed as in [SN05]: it is assumed that — for batches D^* of sufficient size — the style parameter can be reliably inferred using a maximum likelihood approach:

$$s^* = \arg\max_s \left[\prod_{d^* \in D^*} \left(P(d^*) \cdot \prod_{v \in V} P(v|d^*,s)^{n(v,d^*)} \right) \right] \tag{5.9}$$

This leads to an annotation procedure in which the appearance likelihood is computed for each style, and after this style-specific annotation is run for the best style s^*.

5.3.4 Style Variant 2: Appearance-and-Tags

The "appearance-only" style model from Section 5.3.3 makes limited use of style information in the sense that the distribution of tags is assumed to be style-independent. In practice, however, tags may be strongly correlated with style (for example, the tags given to pictures from a New York City sightseeing trip may differ significantly from the ones used for an African safari). To exploit this

information, a second style variant is proposed in which both appearance *and* tags are modeled with style-dependent distributions $P(v|z,s)$ and $P(t|z,s)$, like in Equation (5.1):

$$\begin{aligned} P(t|d,s) &= \sum_{z \in Z} P(t|z,s) \cdot P(z|d) \\ P(v|d,s) &= \sum_{z \in Z} P(v|z,s) \cdot P(z|d) \end{aligned} \tag{5.10}$$

While for the "appearance-only" model, different styles were still connected via the joint use of the tag distribution $P(t|z)$, the new approach leads to a set of entirely decoupled style-specific annotation models. It will be referred to as the "appearance-and-tags" style model in the following. A graphical illustration of the sample generation process can be found in Figure 5.3(c).

Learning and Inference Since styles are now completely decoupled, training simplifies to learning a separate PLSA-based annotation model per style. Similarly to the plain PLSA model in Section 5.3.2 the EM algorithm is used, only that the distributions $P(v|d)$ and $P(t|d)$ are replaced with style-specific equivalents $P(v|d,s)$ and $P(t|d,s)$. These are trained on style-specific training image sets $\{d \,|\, s(d) = s\}$.

For inference, the target style is determined using the same maximum likelihood criterion as for the "appearance-only" style model in Equation (5.9), only that $P(t|z)$ is replaced with its style-specific equivalent $P(t|d,s)$. Again, annotation is carried out using the style-specific model of the best style s^*.

5.4 Experiments

In this section, several experiments are presented in which the proposed style modeling approach for image annotation is evaluated. Tests are run on several datasets compiled from the COREL dataset, which is a standard benchmark for image annotation [CCMV07, DBdFF02, FML04, TL07], and on real-world photo stock downloaded from Flickr.

It has already been mentioned that the proposed framework requires style labels for training. For the COREL dataset, the fact is used that pictures come in *folders* associated with objects ("Mushroom", "Model"), locations ("Africa", "Hong Kong"), or other categories ("Kung Fu", "Reflecting Surfaces"). Pictures are assumed to

Table 5.1: Throughout the experiments in this chapter, different styles are associated with locations and travel scenarios as they might appear in personal collections of holiday snapshots. Datasets were sampled from COREL folders or Flickr groups.

Dataset	Styles Used
COREL-13	Africa Holland Kyoto Monaco Rome Singapore Turkey England Ireland Mexico NY_City Scotland Thailand
COREL-45	Afrcianw Egypt1 Ireland Kyoto Montreal Portugal Singapore Washington Africa Egypt2 Isles1 London Namibia Prague Thailand Yemen Alaska France Jamaica Middle_East New_Guinea Quebec Turkey Zimbabwe Belgium_1 Hawaii Japan1 Mexico New_Zealand Rome Utah Berlin Holland Japan2 Mexico_City NY_City Russia Virginia Devon_UK Hongkong Kenya Monaco Pei Scotland Washington_DC
FLICKR	Africa Alaska Greece Maldive New_York_City Paris Rome Tibet

belong to the same style if they are placed in the same folder. In the Flickr case, images in the same Flickr group are assumed to share the same style. The groups used in the experiments correspond to holiday destinations, like "African Safari" or "New York City Trip".

The proposed style-based model is compared with state-of-the-art results from the literature and with several baselines, all treating images individually.

5.4.1 Setup

This section describes the experimental setup, including datasets, methods tested, and performance measures. Quantitative results will be discussed afterwards.

Datasets Three datasets are used, consisting of different image data, ground truth annotations, and groupings of pictures into style-coherent batches. Two of these sets are subsamples of the COREL dataset [MMMP02], a collection of over 800 photo CDs that is a standard choice for evaluating image retrieval. A 1:1 correspondence between styles and COREL folders is imposed (a complete list of styles/folders can be found in Table 5.1).

COREL-13: 13 folders ($1,300$ images overall) are selected from the COREL dataset corresponding to countries, regions, and cities (for example, "Africa" and "Kyoto"). A vocabulary of 644 tags from the COREL annotations is used.

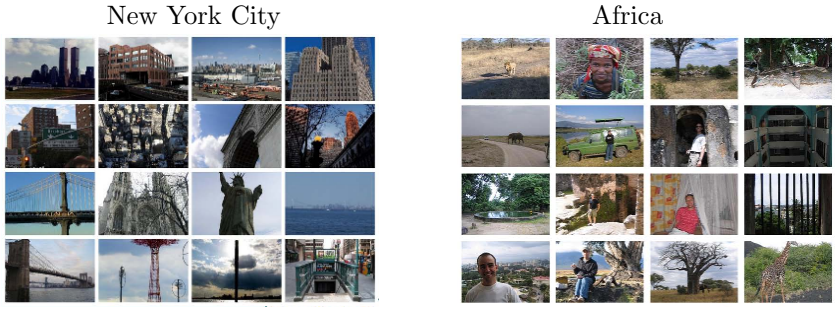


Figure 5.4: Pictures randomly sampled from two styles of the FLICKR dataset. High content variation renders an automatic annotation of these pictures a difficult challenge. Also, it can be seen that appearance differs strongly between styles. Pictures from Flickr.

COREL-45: To compare the performance of style modeling under varying numbers of styles, a dataset similar to COREL-13 is sampled, only that 45 folders (4,500 images overall) are used instead. The tag vocabulary size is 1,257.

Though frequently used, the COREL dataset has been criticized to be oversimplifying, as it contains many near-duplicate images, and words tend to appear in clusters [TL07]. To validate that style modeling works on more challenging and realistic data, the framework is also tested when learning its styles directly from Flickr groups:

FLICKR: This dataset contains 8,000 images downloaded from 8 Flickr groups, each one serving as a style. These styles correspond to travel destinations, i.e. one style contains images taken from New York trips, one shows pictures from an African safari, etc. Please refer to Figure 5.4 for some sample pictures. A vocabulary of 544 terms was created from the most frequent Flickr tags by filtering infrequent or unsuitable ones (like "d40", "2008", or "Olympus").

Features As outlined in Section 5.3, the PLSA annotation approach uses *visual words* as image features. Standard practice for visual word extraction was followed [QMO+07, SREZ05, SZ03]: patches were sampled from each image using a dense regular sampling at several scales, obtaining ca. 4,800 patches per image on average. These were described using SURF features [BTvG06], which were clustered into 2,000 visual words by K-Means (a fast version [Elk03] was used[3]).

[3]available from `http://mloss.org`

Methods The experiments presented in the following include comparisons of various baselines and control runs, indicating how reliably the style of an image batch can be inferred and how style modeling performs compared to an annotation of individual images. Eight different PLSA-based methods were tested, whereas the number of topics was fixed to $|Z| = 20$ (which gave the best results in previous tests):

Baseline, $|Z|$ Topics: This is the plain PLSA annotation model by Monay and Gatica-Perez [MGP04] (Section 5.3.2) using $|Z|$ topics. Images are tagged independently, and style information is discarded.

Baseline, $|Z| \cdot |S|$ Topics: To make sure that potential performance improvements by style are not trivially attributed to a higher number of topics, the baseline was also tested with $|Z| \cdot |S|$ topics (which equals the overall number of topics in the style models).

Appearance-only, Style Assigned: This run uses the "appearance-only" style model from Section 5.3.3. The correct style is assigned for all test images according to ground truth information (which may not be available in practice). This oracle-based control experiment will be used to quantify performance loss due to incorrect style assignment.

Appearance-only, Style by Batch: The same model ("appearance-only" style), but now style is decided automatically based on the whole batch (proposed approach).

Appearance-only, Style by Image: The same model ("appearance-only" style), but the batch size is set to 1, i.e. each image is mapped to a style individually. This method serves as a baseline.

Appearance-and-tags, Style Assigned: The style model from Section 5.3.4. The correct style is assigned according to ground truth (serves as a control experiment similar to "Appearance-only, Style Assigned").

Appearance-and-tags, Style by Batch: The same model ("appearance-and-tags" style), but now style is decided automatically based on the whole batch (proposed approach).

Appearance-and-tags, Style by Image: The same model ("appearance-and-tags"), but now style is assigned for each image individually (serves as a baseline similar to "Appearance-only, Style by Image").

Performance Measures The 4 tags with the highest posteriors (see Equation (5.6)) are selected as annotations for each test image. As a measure of annotation performance, the F-measure (weighted harmonic mean of precision and recall) is used. For each test image d^*, the annotation result $r(d^*)$ is compared to the ground truth tags $gt(d^*)$, obtaining the image-wise precision $\mathcal{P}(d^*)$ and recall $\mathcal{R}(d^*)$:

$$\mathcal{P}(d^*) = \frac{|\, r(d^*) \cap gt(d^*) \,|}{|\, r(d^*) \,|}, \qquad \mathcal{R}(d^*) = \frac{|\, r(d^*) \cap gt(d^*) \,|}{|\, gt(d^*) \,|}$$

By averaging these over all test images, the mean image-wise precision $\bar{\mathcal{P}}$ and recall $\bar{\mathcal{R}}$ are obtained. These are finally combined to the F-measure:

$$\text{F-measure} = \frac{2 \cdot \bar{\mathcal{P}} \cdot \bar{\mathcal{R}}}{(\, \bar{\mathcal{P}} + \bar{\mathcal{R}} \,)}$$

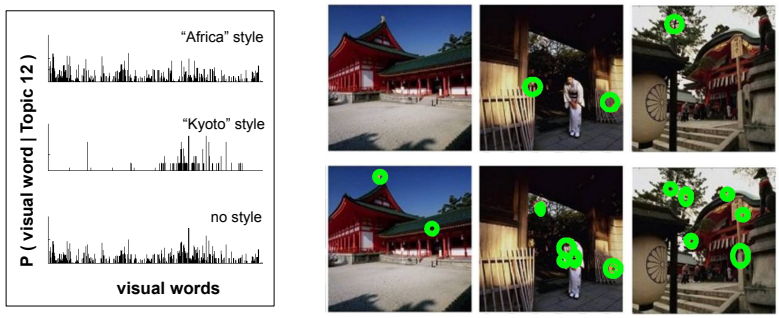
Apart from the final annotation results, we also evaluate the accuracy of the maximum likelihood style decision:

$$\mathcal{A}_{style} = \frac{\text{number image batches assigned the correct style}}{\text{overall number of image batches}}$$

5.4.2 An Illustrative Example of Style Modeling

The experimental evaluation starts with an example that illustrates the effects of style modeling on image annotation. One fundamental requirement for style information to give an improvement is that style-specific models differ from their non-style equivalents. This is demonstrated for a sample topic and the two styles "Africa" and "Kyoto" from the COREL dataset. A topic is chosen (referred to as "Topic No. 12" in the following) whose most frequent tags include the terms "people" and "temple". For this topic, the distribution of visual words $P(v|\text{"Topic 12"})$ is visualized for the baseline model as well as for the two styles $P(v|\text{"Topic 12", "Africa"})$ and $P(v|\text{"Topic 12", "Kyoto"})$ using the "appearance-only" style approach. The result is illustrated in Figure 5.5(a). Obviously, the appearance learned for both styles differs strongly from the one in the global model. In fact, the non-style model can be seen as a mixture of two very different style appearances.

The next question is how well images are fitted to appropriate topics. This is illustrated in Figure 5.5(b). These sample pictures are tagged with "people" and "temple", which are again associated with Topic No. 12. Consequently, a good model should lead to a strong activation of Topic No. 12 in the image. To study whether this is true, the visual words with high topic scores ($P(v|\text{"Topic 12"}) \geq$



(a) (b)

Figure 5.5: (a) The visual word distribution of Topic No. 12 for the styles "Africa" (top), "Kyoto" (center), and without style modeling (bottom). It can be seen that style has a massive influence on topic appearance. (b) The visual words corresponding to Topic No. 12, which is strongly linked to the tags "people" and "temple". Non-style results are in the top row, results for the "Kyoto" style at the bottom. For the style model, more patches can be found that activate the topic, and better tagging results are to be expected (pictures from the COREL Dataset).

0.5) are highlighted for the non-style case (top) and for the correct style "Kyoto" (bottom). While in the baseline model only few patches can be found related to Topic No. 12, for the style approach multiple patches activate Topic No. 12, and a better annotation result can be expected.

5.4.3 Results 1: COREL and Flickr Experiments

Quantitative results for all three datasets are given in Figures 5.6 and 5.8. All results were obtained by averaging the F-measure of annotation performance over multiple runs (20 for COREL-13 and FLICKR, 11 for COREL-45). In each run a random split into 80% training and 20% testing was done, i.e. images were grouped to style-coherent batches of size 20.

Comparison of Style Models: Plot 5.6 provides results for all three benchmarks. Both style extensions are compared with two non-style baselines. It can be seen that the proposed style modeling improves annotation performance significantly: compared to the best baseline, relative improvements between 30.1% (COREL-45) and 146.4% (FLICKR) are measured. All of these are significant according to a paired t-test over all runs (level 99%). Second, "appearance-and-tags"

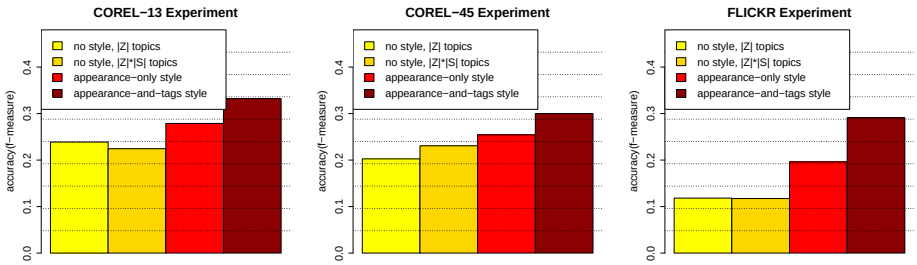


Figure 5.6: Comparing non-style baselines with the two proposed versions of style modeling. It can be seen that style modeling improves performance in all experiments significantly, and that "appearance-and-tags" style outperforms the "appearance-only" style model ($n = 5,200$ [COREL-13], $9,900$ [COREL-45], $32,000$ [FLICKR]).

style outperforms the "appearance-only" style model on all datasets, which can be explained by the fact that the former takes varying tag distributions over styles into account. Again, these relative improvements are significant (paired t-test, level 99%), ranging from 18.0% (COREL-45) to 48.5% (FLICKR).

Comparison with Control Runs: Plot 5.8 compares the proposed method ("appearance-and-tags" style) with two more control runs. One ("style assigned") chooses the correct style for each image batch. The other makes an individual style decision for each image ("style by image").

The first important insight from Figure 5.8 is that batch-wise annotation outperforms an image-wise one in all cases, i.e. using style information drawn from the whole batch of images improves image annotation significantly. This can be seen when comparing the "style by batch" results with the "style by image" ones. Relative performance improvements range from 27.4% (COREL-13) to 101.0% (FLICKR) and are all significant (paired t-test, level 99%). When comparing "style by batch" with the "style assigned" control runs, it can be observed that a moderate relative performance loss occurs due to incorrect style decisions, ranging from 5.0% (COREL-13) to 11.8% (FLICKR). When comparing the COREL-13 and COREL-45 experiment, it can be seen that — when increasing the number of styles — performance loss increases slightly, which can be attributed to the fact that a decision between more styles is more error-prone (accuracy decreases from 85.8% to 66.5%). Overall, the benefits of style modeling decrease slightly when scaling from 13 to 45 styles, but remain significant.

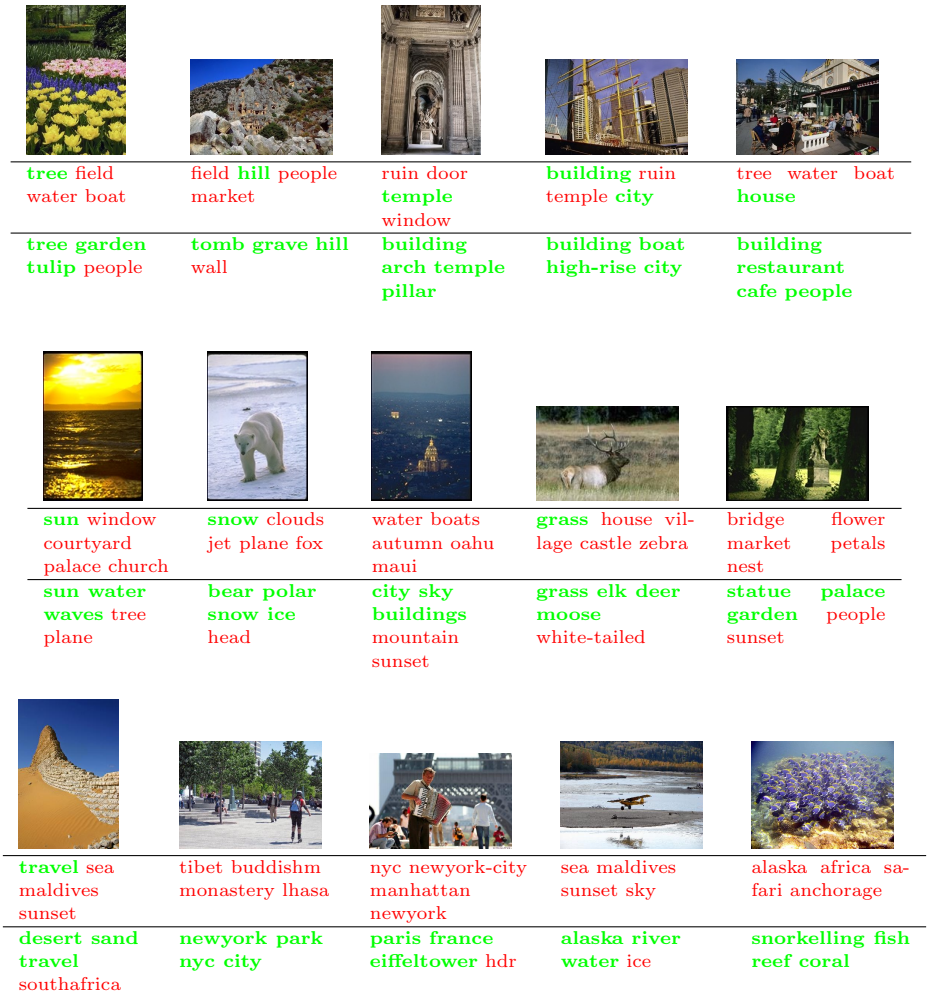


Figure 5.7: Sample annotation results for the COREL-13 dataset (top), the COREL-5K benchmark (center), and the FLICKR dataset (bottom). For all datasets, annotation results without style (top) and with style (bottom) are given. Correct annotations are highlighted in bolt and green, incorrect ones in red. Style modeling improves tagging performance — for example, for the Flickr image at the center of the bottom row, the style-based approach estimates the correct style "Paris" and infers that the Eiffel Tower appears in the background.

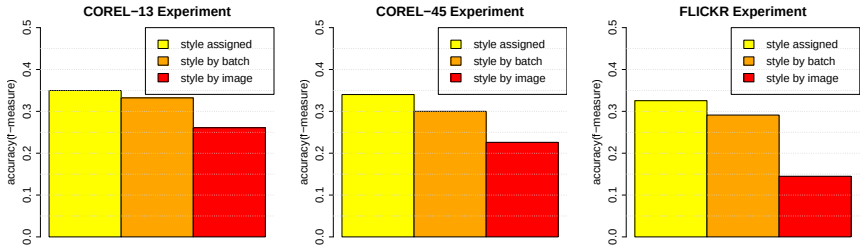
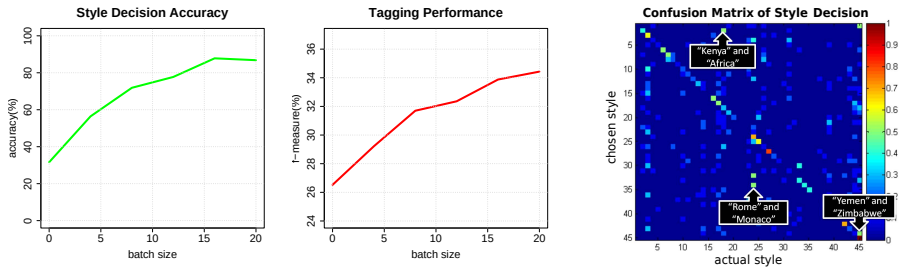


Figure 5.8: Results when using the ground truth style, inferring style using context, and inferring the style separately for each image. The proposed method almost reaches the performance of the oracle-based control run (slight performance loss occurs due to inaccuracies of the automatic style decision), and inferring the style from the batch performs significantly better than deriving the style from each individual image.

Sample Results: Sample annotations are provided in Figure 5.7, where annotations for the non-style baseline are compared with the ones for the "appearance-and-tags" style model. While the baseline provides mostly incorrect annotations, style modeling helps to infer surprisingly good results: for example, for the image showing flowers (top left) the style "Holland" is estimated from the whole image batch, which allows to infer the correct tag "tulip". For a Flickr image showing an accordion player (bottom row, center) the style approach estimates the correct style "Paris" from the batch and is able to infer that the "Eiffel Tower" appears in the background, while the baseline approach assigns the incorrect tag "new york".

Influence of Batch Size The observations made in the last section support the hypothesis that annotation performance is correlated with the test batches' size (for example, "by-batch" runs consistently outperform "by-image" runs). An explanation for this is that the style decision can be made more reliably when based on more images. This is illustrated in Figure 5.9(a), where both the style decision accuracy $\mathcal{A}_{style}$ and the annotation performance are plotted against the test batch size for the COREL-13 dataset (using the "appearance-and-tags" style model and averaging over 10 cross-validation runs). It can be seen that — by increasing batch size from 1 to 20 pictures — the style decision accuracy can be improved significantly from 31.6% to 86.8%, and correspondingly the annotation performance increases from 26.5% to 34.4%. Even for a rather small batch size of 8 images, a relative performance improvement of 20% is achieved (significant according to a paired t-test, level 95%).



(a) (b)

Figure 5.9: (a) The style decision accuracy $\mathcal{A}_{style}$ and the annotation performance both increase with the number of images per style-coherent group. The leftmost point in both plots corresponds to a tagging of individual images. (b) The confusion matrix of style decision on the COREL-45 dataset. The most frequent confusions tend to occur for visually similar styles (like "Kenya" vs "Africa").

Style Confusion Finally, we address the question what styles tend to be confused most often. Figure 5.9(b) illustrates the confusion matrix of style decision for the COREL-45 dataset (using the "appearance-only" style model). An in-depth inspection reveals that the most frequently confused classes are in fact visually similar: For example, "Kenya" and "Africa" are frequently confused (the probability is close to 0.5). Other confusions are "Rome" vs "Monaco", both showing similar classical architecture, or "Yemen" vs "Zimbabwe".

5.4.4 Results 2: COREL-5K Benchmark

In a final experiment, the proposed framework is compared to other methods from the literature. These tests are performed on the COREL-5K benchmark, a frequently used test case for image annotation [CCMV07, DBdFF02, FML04, LMJ04, TL07]. The dataset consists of $5,000$ images from the COREL dataset corresponding to 50 folders of 100 images each. Like in the previous COREL tests, a 1:1 correspondence between styles and folders is imposed. The dataset was split by default into a training set of $4,500$ images (90 images per style) and a test set of 500 images (10 images per style). The batch size was set to 10. The standard tag vocabulary of 374 terms was used, and — similar to all other methods in the literature — the proposed framework returned the top 5 words as annotation results.

The original COREL images were downscaled to a width of 192 pixels. Visual words were extracted by a regular sampling of about $5,400$ patches of side length 12 per image. These were described using DCT coefficients in YUV color space. A K-Means clustering to $2,000$ visual words was done, similar to the one for the previous COREL and FLICKR tests.

The same performance measures are used as in the literature: for each tag t, the per-word precision and per-word recall are measured over all test images d^*:

$$\mathcal{P}(t) = \frac{|\,\{d^* \,|\, t \in gt(d^*) \cap t \in r(d^*)\}\,|}{|\,\{d^* \,|\, t \in r(d^*)\}\,|}, \qquad \mathcal{R}(t) = \frac{|\,\{d^* \,|\, t \in gt(d^*) \cap t \in r(d^*)\}\,|}{|\,\{d^* \,|\, t \in gt(d^*)\}\,|},$$

These values are averaged over all 251 tags occurring in the test set to obtain the mean per-word precision $\bar{\mathcal{P}}$ and recall $\bar{\mathcal{R}}$, which were combined to an F-measure. Also, the number of tags t with $\mathcal{R}(t) > 0$ is reported.

Quantitative results are illustrated in Table 5.2, including a variety of results reported by other researchers: the co-occurrence model by Mori et al. [MTO99], the machine translation model by Duygulu et al. [DBdFF02], two relevance models by Manmatha and co-workers [FML04, LMJ04], supervised multi-class labeling by Carneiro et al. [CCMV07],

and several other annotation models [CR08, TL07].

Our tests also include two baseline approaches: the PLSA model by Monay and Gatica-Perez (Section 5.3.2) and the "appearance-and-tags" style model applied to images individually. Both these baselines do not show a competitive performance (F-measures 5% / 16%). However, by tagging images in style-consistent batches, the proposed approach achieves the best result reported on the COREL-5k benchmark so far, with an outstanding recall of 39%, precision of 25%, and F-measure of 31%. Note that this is achieved by using additional context information, which has not been used on this benchmark so far. Note also that this improvement cannot be attributed to the underlying annotation model (which by itself does not perform competitively), but is clearly due to the exploitation of style. Consequently, it can be expected that other probabilistic annotation models (like Supervised Multi-class Labeling [CCMV07] or Multiple Bernoulli Relevance Modeling [FML04]) could benefit from style modeling in a similar fashion. Overall, these results indicate that context is a valuable information source for autoannotation, and that the proposed integration with style modeling is an appropriate way of using it.

Table 5.2: A comparison of the proposed framework (bottom) with methods from the literature on the COREL-5k benchmark. By making use of context information, the proposed approach achieves the best result reported so far.

Approach	#words with rec.>0	avg. per-word precision	avg. per-word recall	F-measure
co-occurrence [MTO99] (from [CCMV07])	19	0.02	0.03	0.02
Translation [DBdFF02] (from [CCMV07])	49	0.04	0.06	0.05
kernel densities with tag co-occurrence [CR08]	91	0.11	0.13	0.12
SVDCos [TL07]	102	0.15	0.15	0.15
CRM [LMJ04]	107	0.16	0.19	0.17
CSD-Prop [TL07]	130	0.20	0.27	0.23
MBRM [FML04]	122	0.24	0.25	0.24
SML [CCMV07]	137	0.23	0.29	0.26
CSD-SVM [TL07]	127	**0.25**	0.28	0.26
PLSA (no style) [MGP04]	57	0.04	0.09	0.05
PLSA (style by image)	106	0.13	0.23	0.16
PLSA (style by batch) = proposed approach	**141**	**0.25**	**0.39**	**0.31**

5.5 Discussion

In this chapter, a concept detection framework was presented that labels groups of style-coherent images instead of individual ones, such that pictures from the same group can serve as context information for an improved recognition. To make use of such context, an approach based on style modeling was adopted from the domain of optical character recognition. For style learning, category information from the image sharing website Flickr was employed.

Using this framework, significant improvements of up to 100% ($n = 32,000$) have been validated compared to a conventional annotation of individual images. Also, the method achieves the best performance reported so far on the COREL-5K benchmark for image annotation (mean per word precision/recall: 25% / 39%). These results show that context information can give a significant boost to image annotation, and that the proposed style learning from web-based portals like Flickr provides the right way to achieve this.

Several questions demand further investigation along this promising line of research. First, it should be kept in mind that — compared to an image-wise annotation — the proposed method requires some additional information, namely a grouping of test content. Several clues can be used to infer this knowledge, like time stamps and GPS coordinates [CLKH08], the folder structure in a file system, or manual user feedback. A potential problem is that some of these indicators might be unreliable. However, experimental results demonstrate significant improvements even for small groups of images: for batches as small as 8 pictures, a relative performance improvement of 20% was achieved. This shows that context can help even if images are aggregated conservatively into small, reliable groups.

A possible way of improving the system might also be to integrate other image annotation methods. It has been pointed out that the proposed style modeling approach can serve as a wrapper around a variety of models — correspondingly, further performance improvements could be expected by integrating style information with the most successful approaches in this domain, like relevance modeling [FML04] or supervised multi-class labeling [CCMV07].

Other interesting questions are related to a practical, large-scale use of style modeling: how well does the system scale to very large numbers of styles, and how well does it generalize to styles that have not been trained on? While the approach has been validated for up to 45 styles in this chapter, even more might be of interest in practice (ultimately, there are thousands of Flickr groups [NGP08], each one a potential style category). Breaking points of style decision accuracy may occur, and speed issues might have to be overcome, as the complexity of inference is linear in the number of styles. Regarding the generalization to new styles, we can at best expect a mapping of test images to the "most similar" style learned. This may work well in case of a rich collection of styles, but does not have to.

Both these issues might be overcome by integrating other style modeling approaches. One option might be hierarchical Bayesian methods [MB02], which replace a fixed number of discrete styles by modeling style as a continuous parameter sampled from a hyperprior (please refer to Section 5.2 for a brief discussion). It is not straightforward to adapt this approach to the image annotation problem, with its high number of classes and complicated part-based representations. Yet, if such a transfer can be achieved, hierarchical Bayesian methods might be the solution to learn from large style vocabularies and achieve strong generalization capabilities.

Chapter 6

Improving Concept Detection using Motion Segmentation

This chapter proposes a novel combination of concept detection with motion segmentation. The approach is targeted at an improved robustness with respect to clutter: objects are segmented from the background based on their distinctive motion, and recognition is applied on the level of object regions instead of to the whole frame. The following contributions are made, targeted at both an improved motion segmentation and its combination with recognition[1]:

- A novel approach is presented to infer a background motion and region from a given motion field, using a globally optimal branch-and-bound search of parameter space [Bre92]. The method is demonstrated to outperform several local search methods on synthetic and MPEG-4 motion fields.
- A second method is presented that extends a direct motion segmentation [SC06] with statistical color models. This is shown to give improvements over a purely motion-based approach.
- A novel framework for the recognition of objects in video is presented, which combines the above motion segmentation technology with a patch-based object recognition.

[1] This chapter is based on the author's work in [Ulg07, UB08, ULKB07]

- In quantitative experiments on several datasets, the approach is compared to a baseline using unsegmented images. Relative improvements reach 50% (error reduction, $n = 1,584$) for the recognition of specific objects, and 33% (MAP, $n = 4,160$) in a concept detection experiment.

6.1 Introduction

In video retrieval, concepts of interest frequently correspond to physical objects such as buildings, cars, product covers, animals, etc. This is a special case of the more general concept detection setup we have studied so far, and will be the focus of this chapter. The task is closely related to object recognition, which is one of the most intensively studied challenges to computer vision and is concerned with the recognition of specific objects, or — more recently — objects belonging to a certain category such as "car" or "building" [PHSZ07]. Strong connections can also be found with the task of *object matching* [SZ03, SJL+06].

Object matching, object recognition, and object-specific concept detection can all be achieved by relying on the same techniques: correspondences between training views and test images are estimated, obtaining a score that indicates how well both match. In object matching, this score is used to rank images in a database entirely based on visual characteristics, i.e. it is neglected *which* objects are actually present. In object recognition, we use matches for a decision about object presence. Finally, in concept detection, an object-specific score is used for a ranking of content. This chapter is targeted at an improvement of the underlying search for correspondences, such that the proposed approach can be used for object matching as well as object recognition and concept detection. The experiments in this chapter will address the last two challenges, and correspondingly both terms will be used in the following.

It is important to note that all three tasks have strong connections with the segmentation problem, as objects usually occupy only a part of the image or frame. It seems reasonable to assume that recognition becomes significantly simpler if a segmentation of the object from the background is given. However, for still images, such a segmentation turns out to be a difficult challenge: while pictures can be partitioned into regions of coherent appearance, the resulting segments cannot be expected to correspond to meaningful objects or object parts in general. Such a segmentation is called *weak* [SWSJ00]. When it comes to video content, the segmentation problem becomes simpler, as *motion* can serve as an additional clue. In this chapter, we will assume that such a motion-based segmentation can

achieve a *strong* segmentation [SWSJ00], i.e. that we can decompose the scene into a background layer and an object layer (we will focus on the situation of a single foreground object, but approaches to overcome this limitation exist [KTZ04]). Motion segmentation has been demonstrated to achieve this using a variety of approaches [CS05, KRB01, SC06, Tek95, TZ00], even in situations where a strong segmentation cannot be achieved using color and texture alone.

The concern of this chapter is an improved object recognition and object-specific concept detection by using motion segmentation. Motion segmentation will serve as a *filter* for ruling out false positive correspondences with the scene background. The key hypothesis is that this increases the robustness of recognition with respect to clutter.

To realize this idea, we first need to address the segmentation problem. Motion segmentation has been demonstrated to be capable of giving strong scene segmentations in many situations. Yet, it is based on restrictive assumptions like constant pixel intensity or spatial coherence of motion [BA96]. These assumptions are violated in many practical situations, as in case of illumination changes, transparency, or at object boundaries. Therefore, to achieve a robust segmentation, this thesis first presents two improvements and extensions of motion-based segmentation algorithms.

First, an *indirect* approach towards motion segmentation is presented which estimates a motion field and then segments it into coherent regions. This is done by estimating a dominant motion and associated region. In Section 6.4, this idea is followed, and an approach is presented which — in contrast to local search techniques that constitute the state of the art — infers an *optimal* background from a given flow field. This is made feasible by an adaptive search of parameter space using the RAST ("Recognition by Adaptive Subdivision of Transformation Space") algorithm [Bre92]. In experiments on synthetic flow fields and real-world video sequences, the approach is demonstrated to outperform several local search methods in terms of accuracy of segmentation and estimated background motion.

A different, *direct* strategy towards motion segmentation combines motion estimation and segmentation in a joint optimization process, which compensates for the ambiguity of local motion clues. In Section 6.5, an extension of such a *direct* approach with parametric color models is presented which combines motion and color clues in a joint framework. In experiments on a variety of video data, it is demonstrated that this extension reduces segmentation error compared to a purely motion-based approach.

With such motion-based segmentation technologies at hand, the second question addressed in this chapter is whether motion-based segmentation helps to improve object recognition by first segmenting a scene and then applying recognition to the resulting regions. While this idea seems appealing, there are also several arguments against it: first, motion segmentation itself remains a challenging problem, and results can be error-prone and inaccurate. Second, modern patch-based recognition methods by themselves already provide a certain robustness with respect to clutter and give impressive results even for heavily cluttered scenes [FTG06, Low04, SZ06]. Third, in some scenarios, background can be a valuable clue for the presence of an object — for example, the fact that a road is visible hints at the presence of a car.

An answer to the question whether motion segmentation can improve concept detection is given in Section 6.6. A framework for the recognition of moving objects in video is presented that combines motion segmentation with a state-of-the-art patch-based recognition [JDS08a, Low04, MPDB$^+$06, PCI$^+$07]. In the proposed framework, motion segmentation serves as a filter for patches from the background region. This way, incorrect correspondences between object models and scene background are prevented, and the influence of clutter is reduced. Using this framework, the combination of motion segmentation with patch-based recognition is studied. Two quantitative experiments are conducted — one regarding the recognition of specific objects, the other regarding object retrieval in web video databases. Results of both experiments show that, under conditions where motion segmentation can be expected to work, it improves patch-based recognition significantly.

This chapter is organized as follows: after an introduction of some basic terminology and notation (Section 6.2), an overview of motion segmentation and object recognition is given, and previous work targeted at a connection of both fields similar to this chapter is discussed (Section 6.3). After this, the contributions of this thesis with respect to motion-based segmentation are presented (Sections 6.4 and 6.5). Finally, the proposed framework for combining motion-based segmentation with a patch-based object recognition is outlined and validated in two experiments (Section 6.6). A discussion concludes the chapter (Section 6.7).

6.2 Basic Concepts and Notation

In this chapter, a video is viewed as a function over a *volume* of points (x, t), where $t \in \mathbb{R}$ denotes the time (or frame number, as time steps are discrete in practice), and $x \in \mathbb{R}^2$ the spatial location. $I(x, t)$ denotes the intensity of a pixel.

The spatio-temporal derivatives of this intensity are denoted with the gradient $\nabla I(x,t) := (\,\partial I(x,t)/\partial x_1,\, \partial I(x,t)/\partial x_2\,)$ and $I_t := \partial I(x,t)/\partial t$, which denotes the intensity change over time and will be approximated by $I(x,t+1) - I(x,t)$. In some cases, we will also make use of color information, whereas RGB color values will be treated as functions similar to the intensity I. These functions are denoted with $I_R(x,t)$, $I_G(x,t)$, and $I_B(x,t)$.

In most situations, this chapter will focus on motion in a single frame t (or between two frames t and $t+1$, respectively). Therefore, the notation is abbreviated by dropping the time component t: we will replace $I(x,t)$ with $I(x)$, $\nabla I(x,t)$ with $\nabla I(x)$, and even $I_t(x,t)$ with $I_t(x)$.

We will deal with two kinds of motion estimates in this chapter. First, *local* ones, indicating that a feature at position (x,t) moves to $(x+v, t+1)$ in the next frame. $v \in \mathbb{R}^2$ is called a *motion vector*. Usually, local motion is defined at discrete positions $x_1, ..., x_n$ over frame t, obtaining a *motion field*:

$$\{(x_1, v_1), ..., (x_n, v_n)\}$$

where x_i is a position in the frame and v_i a motion vector. Again, the time t is dropped in this notation, as we only focus on a single point in time.

The second kind of motion describes a *global* mapping of positions to motion vectors. This transformation is denoted with $v_\theta(x) : \mathbb{R}^2 \to \mathbb{R}^2$ (where θ is a parameter vector). In contrast to local motion fields, v_θ induces a *dense* estimate of motion over the whole frame, i.e. for any position x, v_θ maps x in frame t to the position $x + v_\theta(x)$ in frame $t+1$.

6.3 State of the Art

As this chapter is targeted at a combination of motion-based segmentation and object recognition, an overview of both fields is given in the following. It should be noted that both tasks have been subject to intensive research since the 1980s (in case of motion analysis [HS80]) or even the 1950s (for object recognition [Mun06]). Therefore, a complete overview of methods is beyond the focus of this section. Instead, the most prominent approaches will be outlined, and pointers to further reading will be provided for the interested reader. We will first address motion-based segmentation (Section 6.3.1) and then object recognition (Section 6.3.2). Finally, methods targeted at combining both fields will be discussed (Section 6.3.3).

6.3.1 Motion-based Segmentation

Motion-based segmentation is targeted at partitioning the frames of a video sequence into regions of coherent motion. The resulting segmentation is of interest in a variety of applications, including obstacle detection [Enk91], video compression [ZL01], the recovery of scene structure [TZ00], or object recognition [KRB01] (which will also be the focus of this chapter).

Let us compare the approach with segmentation in still images, where a grouping is usually done on the basis of low-level clues such as texture or color. Despite strong efforts put into the task, a segmentation of still images remains a difficult challenge, and suffers from the problem that — without prior knowledge of the scene — segmented regions cannot be expected to correspond to meaningful objects [MFTM01]. In contrast to this, a grouping based on motion has a good chance of resulting in regions that correspond to objects (or rigid parts of articulated objects), i.e. a *strong* segmentation is achieved [SWSJ00].

This section reviews related work on motion segmentation in a compact overview (for more information, please refer to the review by Zhang [ZL01], the text book by Tekalp [Tek95], or to a bibliography compiled by Wiskott[2]). Also, it should be noted that background subtraction (or *change detection*) techniques [RAAKR05] — which achieve more reliable segmentations for restricted setups with a static camera – are not covered.

We will first focus on the estimation of motion (which by itself does not include a segmentation of the scene). After this, two general approaches towards motion segmentation will be discussed: feature-based (or *indirect*) methods and direct ones.

Motion Estimation

The estimation of motion in video sequences has been addressed in computer vision since the 1980s, starting with contributions by Horn and Schunck [HS80] and Lucas and Kanade [LK81]. Ideas of how to compute motion are outlined in the following (for an in-depth review, please refer to [BB96]).

Intensity-based Methods: One fundamental assumption of motion estimation is *data preservation* [BA96], which refers to the fact that image intensity stays constant as a feature moves over time. Let us first focus on the estimation of the local motion vector v for a single image feature x in frame t. Then, data

[2]http://itb.biologie.hu-berlin.de/~wiskott/Bibliographies/SegmFromMotion.html

preservation states that:

$$I(x + v, t + 1) = I(x, t).$$

Using a first-order Taylor series expansion, this rewrites as the well-known *optical flow equation* [BB96]:

$$\begin{aligned} I(x,t) + \nabla I(x,t) \cdot v + I_t(x,t) &= I(x,t) \\ \nabla I(x,t) \cdot v + I_t(x,t) &= 0, \end{aligned} \tag{6.1}$$

which provides an elegant way to estimate the motion vector v directly from the spatio-temporal image derivatives. Note, however, that (6.1) is a single equation with two unknowns (namely, $v = (v_x, v_y)$), such that additional constraints are required. This dilemma is commonly referred to as the *aperture problem* [BB96]. As a solution, Horn and Schunck [HS80] introduced an additional regularization term $|\nabla v_x|^2 + |\nabla v_y|^2$ that enforces the estimated flow field to be smooth. Alternatively, Lucas and Kanade [LK81] assumed motion to be approximately constant over a small neighborhood region surrounding x. From pixels in this surrounding, additional constraints can be acquired, leading to a sufficiently determined linear equation system from which v can be estimated in a least squares fashion.

Note that — as the optical flow equation is based on a first-order Taylor series expansion — it gives inaccurate motion estimates in case of strong motion. This can be overcome to some extent using a hierarchical approach, where — starting at a low resolution level — motion is iteratively estimated and used to refine frame alignment [BAHH92]. It has been reported that intensity-based methods can handle motion of up to $10 - 15\%$ of image size this way [IA00].

Parametric Motion: Motion estimation is usually constrained using the fact that features do not move independently but are strongly correlated. One popular strategy is based on the assumption that the flow field follows a low-dimensional parametric form. This is modeled by a mapping of positions x to motion vectors $v_\theta(x)$ (where θ denotes a parameter vector). A variety of such parametric motion models has been suggested (please refer to [BAHH92, Smo01] for an overview). It ranges from complex 3D models to low-dimensional 2D approximations, like the affine transformation:

$$v_\theta(x) = \begin{pmatrix} \theta_1 & \theta_2 \\ \theta_3 & \theta_4 \end{pmatrix} \cdot x + \begin{pmatrix} \theta_5 \\ \theta_6 \end{pmatrix} \tag{6.2}$$

This model gives a sufficiently accurate approximation of the true scene motion if the scene is near-planar and in sufficient distance from the camera. As the number of parameters is as low as 6, these can be estimated reliably even in the presence of

strong noise. This way, parametric approaches increase the robustness of motion estimation significantly.

Feature Tracking: Intensity-based methods as outlined above derive motion directly from the spatio-temporal image derivatives. A second approach is based on a *tracking* of image features over time. One simple technique following this idea is *block matching* [Tou02], which is a key component of modern video encoders: a frame is divided into rectangular blocks, and for each block the most similar region in the previous frame is found under a simple intensity-based block distance measure. Note that motion cannot be estimated reliably in regions showing only line features or no texture at all. This observation motivates a feature selection step previous to motion estimation, such that we focus on areas where accurate estimates can be expected. Corners — i.e. locations with strong image derivatives in either spatial direction — have been pointed out as good choices for this purpose [ST94]. More recently, a variety of other interest point detectors has been proposed [SMB00] that are covariant with respect to strong changes of scale, perspective, and illumination. Based on this development, motion estimation can be performed using a feature matching: features are extracted in two successive frames, and a matching is done based on robust descriptors of local appearance. This approach will be discussed later under the term *feature-based methods*.

It is important to note that all motion estimation strategies above are based on assumptions that are violated in many practical situations. For example, while most methods assume a constant intensity of moving image features, intensity in practice depends on many factors such as the distance and angle of a surface relative to light sources and camera. Another assumption is that motion occurs with spatial coherence, which can be formulated explicitly by smoothness constraints [HS80] or is implicitly assumed by methods such as feature matching. Again, this assumption is violated in practice, particularly at motion boundaries. Heuristics to solve this problem have been proposed based on an inhibition of motion smoothness across image contours [BB96, Ren08], but do not guarantee a reliable solution.

Other fundamental problems are occlusion (i.e., features that are only visible in one of two subsequent frames) and missing image texture (for uniformly colored regions, motion estimates are inherently unreliable). Overall, motion estimation is error-prone and intertwined with motion segmentation, such that any interpretation of motion fields must take outliers and inaccuracies into account.

Feature-based Motion Segmentation

A first category of motion segmentation approaches separates the estimation of motion from its interpretation (i.e., an *indirect* approach is followed). First, local motion is estimated using feature tracking (which is why the approach is also called *feature-based* [TZ00]). Second, the resulting flow field is segmented into regions of coherent motion.

Some methods from this category estimate a parametric 2D background motion (like affine motion, Equation (6.2)) using M-estimators or related robust techniques [FdW05, KCM04, RSSM01, Sib08, Smo01]. Outliers that do not fit this global motion are attributed to foreground objects. While such 2D models can be considered simple and robust alternatives, motion interpretation can also be performed in 3D, which allows for an additional reconstruction of the 3D scene structure. An early approach in this direction has been presented by Adiv [Adi85]: a given motion field is oversegmented into parts corresponding to planar patches under a 3D motion model. These parts are then grouped to obtain a final segmentation.

A similar approach in 2D has been followed by Wang and Adelson [WA93], who decomposed video scenes into several coherently moving *layers*. First, a local intensity-based estimation of optical flow is performed, and after this the motion primitives are alternately segmented and grouped into clusters of coherent affine motion. This clustering is based on a similarity measure in the parameter space of affine transformations, and several such similarity measures have been studied [NWD00]. Another approach by Kühne [KRB01] uses local motion estimation followed by an active contour segmentation. Motion boundaries are iteratively refined, stopping at motion probes that do not fit the global motion estimate. Effectively, this renders motion segmentation an anomaly detection process.

Torr and Zisserman [TZ00] present another system using a matching of corner features. Given the resulting set of point correspondences, a Random Sample Consensus (RANSAC) technique is used to estimate a dominant homographic motion while discarding outliers. It is demonstrated that scene structure can be recovered this way. An evaluation of other 3D motion-based segmentation methods is given by Tron and Vidal [TV07].

Generally, the strengths of the indirect approach lie in the fact that motion estimation can focus on thoroughly selected features. Over the last years, techniques have been developed to make feature detection and description robust to photometric and geometric variations [Mik03, SMB00], such that — by focusing on thoroughly selected image areas — sparse but reliable motion estimates can

be obtained, even under significant scene motion (in fact, feature matching is also applied for *wide baseline* stereo matching [MCMP02]). The problem of sparseness can be overcome by a postprocessing that infers dense flow representations — for example, Belongie et al. [WAB03] start from a feature point matching and then apply a graph cut technique to assign pixels to coherent motion layers. For a detailed discussion of feature-based methods, please refer to [TZ00].

Direct Motion Segmentation

While supporters of feature-based methods argue that a selection of well-trackable features leads to a more robust motion estimation [TZ00], a problem is that — as motion estimation and interpretation are separated — a recovery from errors made in the motion estimation step is difficult. This is particularly true as the estimation of motion is intertwined with knowledge of region boundaries, which is why motion segmentation is also called a "chicken-egg problem" [CS05]. This insight motivates *direct methods*, which estimate motion boundaries and motion itself in a joint process operating directly on image intensities. A variety of methods follows this idea, most of them based on the optical flow equation (6.1): we expect for each pixel x that $\nabla I(x) \cdot v + I_t(x) \approx 0$, and the quantity $\nabla I(x) \cdot v + I_t(x)$ is called the *optical flow error*. A minimization of this error is the key idea of direct *dominant motion estimation* methods: it is assumed that the majority of the frame (typically, the background region) follows a parametric global motion. This is estimated together with the associated region by minimizing flow error in a robust fitting process [IRP94]. Black and Anandan [BA96] propose to replace quadratic flow error as used previously [HS80] with error measures from robust statistics [Hub74], which are less prone to the influence of outliers. A segmentation can be obtained by recursively removing already explained regions and repeating dominant motion estimation until the whole frame is fitted.

Another category of direct methods views motion segmentation as a parameter estimation problem in a probabilistic setting. Assuming image data I to be given in form of spatio-temporal derivatives, a joint estimation of the motion field V and label field L (representing a segmentation) is obtained using a Bayesian formulation:

$$\hat{V}, \hat{L} = \arg\max_{V,L} P(V, L|I) = \arg\max_{V,L} P(I|V, L) \cdot P(V|L) \cdot P(L) \qquad (6.3)$$

If taking the logarithm, such formulations lead to the minimization of energy functions consisting of additive terms. A *data fit term* represents $-\log P(I|V, L)$, the negative log-likelihood of the image data given the motion field. This is balanced

with regularization terms that typically penalize the length of motion boundaries. Bouthemy and François [BF93] were the first to propose such a formulation. For the data fit term, motion within each region is assumed to follow a parametric 2D model, and a Gaussian distribution of optical flow error is assumed. for the regularization terms, an Ising model is proposed, which effectively leads to a Markov Random Field for motion segmentation. Similar formulations were proposed later by Cremers and co-workers, where a quality criterion similar to Equation (6.3) is optimized in an active contour framework [CS05] or using a graph cut approach [SC06]: starting from an initial guess of the segmentation, alternately parametric motion estimates for each region are obtained using least squares, and based on these parameters pixels are re-assigned to appropriate regions. If using graph cut optimization for this assignment, segmentation can be carried out in near-realtime [SC06] or even in real-time (if combined with a multi-resolution approach [VGWK08]).

Other methods use similar probabilistic formulations as Equation (6.3) and propose the EM algorithm for parameter estimation. Jepson and Black [JB93] present a model based on a *mixture* of parametric motions (including an outlier process). In an EM fashion, soft assignments of pixels to these motions are alternated with parameter updates. Weiss [Wei97] proposed a non-parametric approach, where motion is enforced to be *smooth* using the prior term $P(V|L)$. While previous formulations such as the one by Horn and Schunck [HS80] did not take motion discontinuities into account and enforced smoothness over the whole frame, Weiss restricted this constraint to apply only within the segmented regions. Again, optimization is based on the EM algorithm.

It should be kept in mind that most direct methods are based on the optical flow equation, which itself relies on the pixel constancy assumption. Violations of this requirement (for example in case of illumination changes) have been addressed using heuristic normalization techniques [IA00], but remain a serious challenge. On the other hand, the strengths of direct methods lie in the fact that segmentation and motion estimation are coupled in a joint process, that highly accurate motion estimates in the subpixel range are obtained (which is mandatory for some applications such as superresolution and mosaicing), and that the recovered motion segmentations are inherently dense. For a more detailed discussion, please refer to Irani's and Anandan's work [IA00].

6.3.2 Patch-based Object Recognition

The recognition of objects has been in the focus of computer vision research since the 1950s [Mun06]. Though an accurate recognition of a large number of generic objects remains beyond the capabilities of state-of-the-art systems, intensive research has lead to an increasing robustness and broader applicability. The focus of this section will be on *patch-based* methods, which have become increasingly popular over the last years, and have also been implemented in commercial systems [MPDB+06][3]. Yet, it should be kept in mind that a plethora of other approaches towards object recognition exist, the most important ones including global appearance-based methods (which are extensively discussed in a recent overview by Roth [Rot08]), and shape-based methods (which have been surveyed by Pope [Pop94]). For a historical overview of the field, please refer to Mundy's article [Mun06].

Object recognition poses a difficult challenge, since robustness must be achieved with respect to variations of illumination, viewpoint, rotation, and pose. Further difficulties are clutter, intra-class variation of object categories, and the effort associated with training data acquisition. In different use cases, a different emphasis may be put on each of these challenges, leading to different requirements to recognition systems in practice. For example, the best choice of features may depend on properties of the objects to be recognized (different representations are helpful for heavily textured objects and for untextured ones).

In this chapter, the focus will be on object retrieval in video databases, an extremely difficult challenge that can be characterized by strong intra-class variation and significant clutter. Also, a learning of object models should take place on similarly challenging video datasets labeled with objects of interest. In this scenario, patch-based approaches have been demonstrated to be an appropriate choice [EVGW+08, JDS08a]. They are adopted in this chapter, and are correspondingly in the focus of the following overview.

Patch-based methods are based on the observation that the image parts related to an object must be identified and separated from the background during the recognition process. The fundamental prerequisite for this is that — in contrast to global appearance-based methods [NaSN96] — an image is described as collections of local parts (or *patches*, respectively). By combining this approach with robust methods for a detection and description of interest regions, a high robustness has been achieved with respect to perspective changes, deformations, clutter, and partial occlusion. This section provides an overview of the most im-

[3] http://www.kooaba.com/technology/

portant trends in patch-based recognition. For further reading, a variety of web resources exist which provide an introduction to the field, like the tutorial on local features by Tuytelaars at ECCV 2006[4], a tutorial on visual recognition given by Leibe and Grauman[5], and a short-course by Fei-Fei et al. at ICCV 2005[6]. A comprehensive overview of different approaches is also given in the textbook by Ponce et al. [PHSZ07]. This section will first address the patch-based recognition of specific objects and after this the recognition of object categories.

Recognition of Specific Objects

Patch-based methods targeted at a recognition of specific objects usually cast a *matching* problem on the basis of local features, which is conducted in four major steps. First, regions of interest are detected. Second, the appearance of each region is described by a numerical feature. Third, these descriptors are matched with patches from training views of the object, obtaining a (potentially error-prone) set of correspondences. In a final step, this set of matches is *refined* — usually by making use of the spatial constellation of patches — and a decision of object presence is made, typically by choosing the class that maximizes the number of correspondences. In the following, these processing steps will be discussed in detail:

- **Patch Selection**: Region selection can simply be performed by sampling patches over random or regular positions and scales. While this usually produces lots of patches (which has been shown to increase system robustness in certain cases [NJT06]), a more efficient way is to focus on *interest points* allowing a repeatable detection. Typical examples are *corners*, for which the Harris detector offers a popular choice [HS88]. More recently, scale-invariant detectors for corners [MS04] and blobs [BTvG06, Low99] have been proposed. Other methods identify so-called *maximally-stable extremal regions* with a strong contrast to their surrounding [MCMP02], or detect maxima of a saliency measure [KB01]. An evaluation of these methods in terms of detection repeatability has been conducted by Schmid et al. [SMB00]. For an extensive overview, please refer to the survey by Roth [Rot08].

- **Patch Description**: Local features are used to describe the distinct appearance of a patch while achieving robustness to changes of rotation and

[4]http://homes.esat.kuleuven.be/ tuytelaa/ECCV06tutorial.html
[5]http://www.vision.ee.ethz.ch/ bleibe/teaching/tutorial-aaai08/
[6]http://people.csail.mit.edu/torralba/shortCourseRLOC/index.html

illumination. Like for the region selection step, a variety of methods has been proposed. Many of these — like the popular SIFT descriptor [Low99] — capture localized statistics over gradient directions (which are inherently robust with respect to illumination changes) [BTvG06, BMP02]. Invariance to rotation can be achieved by a normalization to a characteristic angle [Low99] or by aggregating information from multiple angles in the same histogram bin [JH99]. For an overview and a performance study, please refer to [Mik03, Rot08].

- **Matching**: In this step, correspondences are found between patches in the test image and in training views. This is often done using a nearest neighbor matching of patch descriptors: given a patch from a test image, the most similar one from all training views is found and declared to be a match. This is called a *full patch search* in the following. More efficient alternatives are based on a vector quantization of patches to *visual words* [SZ06], such that matching is done with a significantly lower number of patch prototypes. This way, an efficient search is possible even in large image databases, particularly when using hierarchical extensions of visual word vocabularies [JDS08b, NS06, PCI+07].

- **Refinement**: Frequently, a refinement of error-prone correspondences is done based on the spatial arrangement of patches. This is based on the assumption that object features come with a restricted spatial constellation, which does not hold for false positive matches. One strategy models a global transformation from patch positions in training views to positions in the test image. This transformation can be estimated using RANSAC [FB81], a combination of the generalized Hough transform and robust least squares [Low04], or the RAST algorithm [KDB07]. Alternatively, training and test image can be viewed as a stereo pair, and epipolar geometry can be employed [FP02]. While all these methods define a *global* relation between model and image features, others mine the match set for *locally* correct constellations of patches. In their VideoGoogle system, Sivic and Zisserman and Ferrari et al. [SZ04, FTG06] detect consistent configurations of patches by analyzing local patch neighborhoods. Jegou et al. require valid matches to show a consistent orientation and scale, but neglect their spatial constellation [JDS08a]. Finally, another alternative is to discretize the position of local features into bins [LSP06].

It should be pointed out that — while multiple alternatives exist regarding the above design choices — there is no agreement on an optimal configuration. For example, evaluation efforts for interest point detection and description show very different results on different datasets [FB04, Mik03, MP05, SMB00]. Yet, patch-based approaches in general have been demonstrated to be very successful when it comes to the recognition of specific objects.

Object Category Recognition

The recognition of *object categories* (like cats, airplanes, buildings, etc) is usually considered a much more challenging problem compared to the recognition of specific objects. This is because instances within a category may vary significantly in their appearance, such that recognition systems need to distinguish what variations are inherent to the class and which ones make a distinction to others. Yet, the field has experienced a boost over the last years due to intensive research [PHSZ07, EVGW+08]. Just like for the recognition of individual objects, the patch-based paradigm plays an important role: like objects, object categories are often described as collections of local *parts* associated with patches.

Early examples of patch-based object category recognition are the Constellation Model [BWP00, FPZ03, FPZ05] and the Implicit Shape Model [LS07], which both combine probabilistic distributions for the appearance of object parts (like the handles and wheels or a motorbike) with a model for the spatial arrangement of parts in the image plane. In case of the constellation model, appearance is modeled by Gaussians in the space of local descriptors, and pairwise terms define the relative spatial arrangement of object parts. Recognition is carried out by evaluating all possible constellations, which corresponds to a marginalization over latent correspondences. A similar approach is followed by Leibe's and Schiele's Implicit Shape Model [LS07], though some different design choices are made: patch appearance is discretized to visual words, and the spatial constellation of patches is modeled relative to the object center, such that the Generalized Hough Transform can be used for an efficient localization.

A variety of approaches has been proposed using similar ideas [HL04] or suggesting hierarchical extensions [OB07, UVNS02]. Most of these methods focus on object views that are restricted to a consistent viewpoint, and the object is also required to be prominent compared to its background. Correspondingly, these approaches achieve a good performance on oversimplifying datasets such as the Caltech image collection [PBE+06], which "invite over-optimization to trivial regularities" [PCD08].

To investigate more realistic situations (in which occlusion, extreme clutter, and general viewpoint changes pose additional challenges), object category recognition has also been evaluated on web photos downloaded from Flickr. In this benchmark, the PASCAL Visual Object Challenge [EVGW+08], a fairly simple approach de-emphasizing patch position was found to give the best results, which discretizes features into visual words and stores them in histograms. The spatial constellation of patches is neglected entirely or used in a very limited fashion, for example by dividing the image into subwindows and storing one histogram per window [LSP06]. The resulting feature vector is fed to a *Support Vector Machine* (SVM) classifier [SS01]. This approach has been found successful though it is fairly simple: no segmentation of the object from the background is involved, and neither is an estimation of object pose. It has therefore been argued that there might be more appropriate models for object recognition, but that the PASCAL datasets might be "too difficult for any model to gain traction [...], giving little insight on which approaches are most promising" [PCD08].

6.3.3 Combining Object Recognition and Segmentation

Obviously, the problems of recognizing objects and segmenting them from the background are strongly intertwined, and it seems reasonable to assume that solutions to one task can help solving the other: on the one hand, a segmentation of the object from the background may simplify the recognition problem. On the other hand, an object model used for recognition can serve as a clue for segmentation. Therefore, this section discusses previous work targeted at combining segmentation and recognition.

Top-Down Segmentation: A number of approaches has been proposed for a joint segmentation and object recognition in still images. These methods can be subsumed under the term "top-down segmentation" [BU02], as high-level knowledge of object shape and appearance complements low-level image clues during the segmentation process.

A first category of methods follows this approach for specific objects. Simon and Seitz [SS07] and Ferrari et al. [FTG06] propose to solve recognition through the exploration of a dense feature correspondence field. Starting from an initial set of sparse matches (which can be found reliably even for wide variations of viewpoint), a dense covering of the image is achieved by iteratively exploring the surrounding of matches for further correspondences. This approach gives a dense segmentation and has been reported to lead to an improved recognition, but is also computationally expensive [FTG06].

Beyond this, several approaches have been suggested dealing with instances of an object *category*. Borenstein and Ullman [BU02] and Leibe and Schiele [LLS04] present models that achieve a category-specific object segmentation, which are based on a matching of image parts with patches from segmented training views. While Borenstein and Ullman start from a few reliable matches and expand the match set successively, Leibe and Schiele use the Hough transform as a global approach. Similar to a jigsaw puzzle, the resulting patch-level segmentations are stitched to a global, object specific segmentation.

Todorovic and Ahuja [TA06] follow a graph-based approach. Given many views of object instances, a hierarchical bottom-up segmentation of each image is performed, obtaining a *segmentation tree*. Then, an object model is inferred as a subtree common to all images. Other approaches are targeted at an object-specific extension of Conditional Random Fields (CRFs), which are frequently used models for segmentation. In conventional CRFs, "data" terms express the fit of image regions to foreground and background, and are combined with "smoothness" terms enforcing a spatial continuity of segmentation. This formulation is now extended with "object terms" penalizing deviations from an object category model. For example, the *Layout Consistency CRFs* by Winn and Shotton [WS06] model the appearance of object parts via detectors based on discriminative classifiers. Kumar et al. [KTZ04] present a model that integrates appearance terms with part-level penalty terms for invalid spatial constellations. Other approaches [KTZ05b, WJ05] use pixel-level penalty terms, such that segmentations are modeled as deformed variations of a latent prototypical object shape.

All these models have in common that object-specific shape and appearance terms are used to guide segmentation. Usually, an interleaved fitting process is applied: alternately, the object model is mapped into the image and image regions are fitted to this model. Differences cannot only be found in terms of the exact prior terms, but also in the supervision of learning: while some methods require manually segmented object views for training [BU02, LLS04], others infer a motion-based segmentation from training videos [KTZ04, KTZ05b] or learn from entirely unsegmented images [TA06, WJ05].

This work has led to first promising results: it has been shown that more accurate and meaningful segmentations can be acquired compared to a purely bottom-up approaches, and that a recognition of object categories is possible as well. Two aspects have not been addressed so far. First, all methods mentioned above have only been applied to images where objects are prominent and are viewed from the same canonical pose (often, side views). It is controversial in how far such input data represents natural scenes "well" [PCD08], and — though progress

may be seen in this direction in the near future – it currently remains unclear how well these methods generalize to arbitrary 3D viewpoint changes and general objects. Second, evaluations focus on the segmentation aspect in most cases, and — while a segmentation of objects has a value in itself for other applications — improvements over competitive no-segmentation baselines for recognition have not been validated satisfyingly. In this chapter, both these limitations are addressed: a much simpler recognition approach is followed based on patch appearance only (and not on object pose), which can be applied even in case of wide viewpoint variations. Also, a quantitative evaluation will be presented in which the proposed approach is compared to competitive standard baselines free of segmentation.

Object Retrieval: The combination with segmentation has been viewed from a different perspective in object matching [SZ03, SJL+06]. Here, one or more query views of an object are given, and a dataset of images or videos is to be searched for other views of the same object. Similar to the recognition problem, it seems reasonable to assume that a segmentation simplifies the retrieval task, as appropriate similarity measures can be found more easily in the absence of clutter. Smeaton et al. [SJL+06] present a user study in which test persons were asked to retrieve objects such as "historical buildings" or "palm trees" from a video database. This dataset was manually segmented previous to the study, and test persons were offered the possibility to segment query objects previous to retrieval. It was found that users made intensive use of this functionality, and that this did indeed lead to a more efficient search.

In practice, however, a segmentation of database content may be infeasible, such that real-world prototypes for object-based retrieval deal with segmentation in different ways: patch-based approaches like the VideoGoogle system [JDS08a, PCI+07, SZ06] usually omit it and rely on a filtering of correspondences to achieve robustness to clutter. Wang et al. perform an unsupervised clustering of interest points based on their position [WLT08] — similar to the approach followed in this chapter, a grouping of patches to supposed objects is performed. The approach proposed here, however, employs motion as a strong additional clue to guide this grouping and to obtain more exact object regions.

Motion Segmentation and Recognition: Only a few previous contributions can be found targeted at a combination of motion-based segmentation and object recognition. Kühne et al. [KRB01] perform a motion segmentation and feed the resulting foreground regions to a shape-based classification. While the approach indicates the capability of motion segmentation to accurately segment objects from the background, recognition experiments are only conducted on very small datasets and do not include comparisons with competitive baselines free of segmentation.

Rothganger et al. [RLSP06, RLSP07] present an object recognition system based on 3D patch models. Using *structure from motion* [TM93], patches extracted from a video scene are segmented into rigidly moving components corresponding to objects. From this information, object models are constructed that capture both the 3D pose and the appearance of each patch. During the recognition process, these models can be matched with images or other 3D models, resulting in a recognition process that combines geometry and appearance. It is demonstrated that this approach outperforms a plain matching of patches in 2D [RLSP06]. The framework proposed in this chapter makes use of motion information in a similar way to segment the scene into rigidly moving components and thus erases false positives from the match set. However, instead of a full 3D reconstruction, it uses 2D segmentation into *layers* of coherent motion. This can be seen as a simpler (and computationally much more efficient) alternative.

6.4 Global Motion Estimation by Adaptive Search of Transformation Space

This section addresses the problem of estimating a parametric global motion for scenes including moving foreground objects. As has been outlined in Section 6.3.1, approaches for solving this task can generally be subdivided into two categories: direct methods, which estimate a global motion directly from image intensities, and feature-based (or *indirect*) ones, which first estimate a motion field and then infer a global parametric motion from it. It has been mentioned previously that strong arguments in favor of either approach can be found [IA00, TZ00].

In this section, an indirect approach will be followed. The focus is not on the motion estimation step, i.e. a motion field is assumed to be given. As common for indirect methods, the majority of the motion field is assumed to follow a global parametric motion. The parameters of this motion are unknown, and global motion estimation is the task of inferring them. As the estimation of background motion also involves the estimation of an associated region, we can apply global motion estimation recursively and use it for segmentation, whereas all regions that have been found to fit the global motion are removed. This process is repeated until the whole frame is explained, i.e. first the background is segmented, then a foreground object, then a second one, etc.

What makes global motion estimation difficult is the presence of noise and outliers, which can be caused by moving foreground objects or by errors in the motion estimation process. Obviously, we face a “chicken-egg” problem. We want

to infer a parametric global motion from the background region. On the other hand, this region must be inferred from the (unknown) background motion. Standard approaches for solving this problem rely on a local search in parameter space. Examples are the RANSAC algorithm [FB81], which iteratively tests motion estimates derived from subsamples of the motion field, or robust least squares methods (or M-estimators, respectively) [Hub74, Ch. 4] [Smo01], which alternately refine motion estimates and reject outliers. These methods can get caught in local minima (as in case of robust least squares) or give a good solution only with a certain probability (as RANSAC).

To overcome this problem, this section is targeted at a full search of parameter space giving an *optimal* result. One way to achieve this is the Hough transform [Bal81]: the parameter space is divided into bins, motion probes cast votes associated with parameter regions, and the bin with most votes is returned as a result. While this does conceptually provide a full search of parameter space, no rigorous solution is provided to cope with measurement noise, and the approach gets inefficient for a small bin size [Bre93]. To some extent, such problems can be overcome by combining the Hough transform with a refinement using local search [Low04]. This, however, is again not guaranteed reach a global optimum.

In this section, it is shown that an optimal solution to the global motion estimation problem can be found using an adaptive branch-and-bound search of transformation space. This approach is called RAST ("Recognition by Adaptive Search of Transformation Space"). It has been developed by Breuel [Bre92] and has been previously applied for the fitting of geometric primitives like lines [Bre96] and rectangles [Bre03a], for object detection [KDB07], and for locating modes in kernel densities [Wir09]. The work presented in this chapter is the first one using RAST for global motion estimation.

The section starts with a probabilistic formulation of global motion estimation as a maximum-likelihood parameter estimation problem. Also, the RAST algorithm is described (Section 6.4.1). Next, this model is extended such that the spatial coherence of segmented regions is taken into account (Section 6.4.2). Finally, experiments on synthetic motion fields and MPEG-4 motion vectors from real-world video sequences are presented in Section 6.4.3. In these tests, it is demonstrated that the proposed optimization does in fact give a superior accuracy in terms of motion estimates and segmentation compared to local search techniques.

6.4.1 Approach 1: Maximum-likelihood

We assume a motion field $D = \{(x_1, v_1), ..., (x_n, v_n)\}$ to be given such that each entry consists of a 2D position x_i and a 2D motion vector v_i, indicating that a feature at position x_i in the first frame moves to $x_i + v_i$ in the second one. D can represent an optical flow field defined at regular positions or sparse tracked point features. Restrictions to motion probes arranged on a regular grid will be made later.

The key assumption of global motion estimation is that the motion field D can be approximated well by a parametric transformation. This transformation is denoted with $v_\theta : \mathbb{R}^2 \rightarrow \mathbb{R}^2$. It maps positions x to motion vectors $v_\theta(x)$. In this section, the 4-dimensional *similarity transform* model with parameters $\theta = (\alpha, \sigma, c_x, c_y)$ will be used, which is based on a 2D rotation by an angle α and a scaling σ, followed by a translation (c_x, c_y):

$$v_\theta(x_i) = \begin{pmatrix} \sigma \cdot cos\ \alpha - 1 & -\sigma \cdot sin\ \alpha \\ \sigma \cdot sin\ \alpha & \sigma \cdot cos\ \alpha - 1 \end{pmatrix} \cdot x_i + \begin{pmatrix} c_x \\ c_y \end{pmatrix} \tag{6.4}$$

Global motion estimation is the task of inferring the parameters θ such that v_θ fits D "well", i.e. $v_i \approx v_\theta(x_i)$. This goodness-of-fit can be expressed using a maximum-likelihood formulation:

$$\hat{\theta} = \arg\max_\theta \quad p(\, D \,|\, \theta \,) \tag{6.5}$$

We assume the motion probes $(x_i, v_i) \in D$ to be independent and drawn from a distribution $p(x_i, v_i|\theta)$. A simple choice for this distribution would be to model inaccuracies of motion estimation with isotropic Gaussian noise [TZ00] such that $p^*(x_i, v_i|\theta) = \mathcal{N}(v_i; v_\theta(x_i), \sigma^2 I)$. In practical flow fields, however, *outliers* occur. These can be caused by failures of the motion estimation process or by foreground objects moving into a different direction. Since we do not have prior information on the motion of such objects, we assume a uniform distribution within a reasonable range:

$$p'(x_i, v_i|\theta) = \begin{cases} c & ||v_i||_2 < \frac{1}{\sqrt{\pi c}} \\ 0 & \text{else} \end{cases}$$

Assuming that in practice no outlier motion occurs outside this range $||v_i||_2 < \frac{1}{\sqrt{\pi c}}$, we simply write $p'(x_i, v_i|\theta) = c$. If combining global motion p^* and outlier process p', we obtain the following motion model:

$$p(x_i, v_i|\theta) \propto\ \max\left\{ \mathcal{N}(v_i; v_\theta(x_i), \sigma^2 I),\ c \right\}. \tag{6.6}$$

This implies that if a motion vector v_i deviates too far from its expected background motion $v_\theta(x_i)$, it is considered an outlier. By inserting this term into the overall likelihood $p(D|\theta)$, we obtain the following optimality criterion:

$$\begin{aligned}
\hat{\theta} = \arg\max_\theta p(D|\theta) &= \arg\max_\theta \prod_{i=1}^{n} p(x_i, v_i|\theta) \\
&= \arg\max_\theta \sum_{i=1}^{n} \log p(x_i, v_i|\theta) \\
&= \arg\max_\theta \sum_{i=1}^{n} \max\left(\log \frac{1}{2\pi\sigma^2} - \frac{1}{2\sigma^2}(v_i - v_\theta(x_i))^2, \log c\right) \\
&= \arg\max_\theta \sum_{i=1}^{n} \max\left(1 - \frac{(v_i - v_\theta(x_i))^2}{\epsilon^2}, 0\right).
\end{aligned}$$

with $\epsilon^2 := 2\sigma^2 \log \frac{1}{2\pi\sigma^2 c}$. Correspondingly, maximizing the likelihood $p(D|\theta)$ is equivalent to maximizing the quality function:

$$Q_1(\theta) = \sum_{i=1}^{n} \underbrace{\max\left(1 - \frac{(v_i - v_\theta(x_i))^2}{\epsilon^2}, 0\right)}_{s(x_i, v_i; \theta) \in [0,1]}. \tag{6.7}$$

This optimality criterion has been used for global motion estimation before (for an example, see [Smo01, Ch. 4]). It is an M-estimator [Hub74], i.e. it consists of local truncated quality contributions that can be interpreted as negative log-likelihoods of the data points. These contributions are denoted with $s(x_i, v_i; \theta)$, and are referred to as the *support* of a local flow probe (x_i, v_i) for a global motion θ. $s(x_i, v_i; \theta)$ is zero exactly if v_i deviates further than ϵ from the model motion $v_\theta(x_i)$ (in this case, we call the probe an *outlier*). The parameter ϵ determines the allowed deviation of a motion sample from the global motion v_θ and is set manually in practice.

Optimization using RAST To optimize the (highly non-convex) target function Q_1, a full search of parameter space is conducted which — in contrast to local search techniques — is guaranteed to find the global optimum. The approach is called RAST (Recognition by Adaptive Search of Transformation space)[7]. We first

[7] an open source implementation by Christoph Lampert was used: http://christoph.lampert.googlepages.com/work/software

Algorithm 2 The RAST algorithm for global motion estimation.

given: the parameter space $\mathbf{\Theta}$ and a motion field D
insert $\mathbf{\Theta}$ into the priority queue q (which is sorted by $\mathcal{U}_1$)
repeat
  extract the first element Θ from q
  split Θ into substates Θ_0 and Θ_1
  compute $\mathcal{U}_1(\Theta_0)$ and $\mathcal{U}_1(\Theta_1)$ using interval arithmetic
  insert Θ_0, Θ_1 into q
until Θ is small enough
return Θ

define a parameter range $\mathbf{\Theta}$ to be searched (which is chosen to include all global motions expected). A hyperrectangle is chosen, i.e. $\mathbf{\Theta} = [\theta_1^0, \theta_1^1] \times ... \times [\theta_d^0, \theta_d^1]$. The strategy of RAST is to find the global maximum $\hat{\theta} = \max_{\theta \in \mathbf{\Theta}} Q_1(\theta)$ using a branch-and-bound search over parameter space. Subregions are searched recursively until we converge in a sufficiently small result region. To choose subregions for the algorithm to focus on, an upper bound $\mathcal{U}_1$ for the maximum value of Q_1 within a subregion $\Theta \subseteq \mathbf{\Theta}$ is computed.

$$\mathcal{U}_1(\Theta) \geq \max\{\, Q_1(\theta) \mid \theta \in \Theta\}. \tag{6.8}$$

The higher this bound, the more promising a subregion is considered to be. Based on this assumption, the algorithm works as described in the following (a pseudocode listing is given in Algorithm 2): starting with the whole parameter space $\mathbf{\Theta}$, we iteratively identify the region $\Theta \subseteq \mathbf{\Theta}$ with maximum upper bound $\mathcal{U}_1(\Theta)$. This region is investigated in more detail: a threshold t and a dimension d^* are picked, and Θ is split into two parts $\Theta_0 = \{\theta \in \Theta \,|\, \theta_{d^*} < t\}$ and $\Theta_1 = \Theta \setminus \Theta^0$. For either of these parts, the upper bound $\mathcal{U}_1$ is computed, and the whole process of selecting a promising subregion and splitting is repeated.

The upper bound $\mathcal{U}_1$ is computed using *interval arithmetic* [Bre03b]: given input values $x \in [a, b]$ and $y \in [c, d]$ and an arithmetic operation $\circ \in \{+, -, \cdot, /\}$, interval arithmetic returns the minimal interval known to contain $x \circ y$. This interval is denoted with $[a, b] \circ [c, d]$. As the quality function Q_1 to be optimized is computed using basic arithmetic operations on values known to come from a parameter range Θ, interval arithmetic can provide an interval containing all possible values of Q_1 within Θ. This interval is denoted with $[Q_1^0(\Theta), Q_1^1(\Theta)]$, and it holds that:

$$Q_1(\theta) \in [Q_1^0(\Theta), Q_1^1(\Theta)] \quad \forall\, \theta \in \Theta.$$

Correspondingly, we set $\mathcal{U}_1 := Q_1^1(\Theta)$. One important aspect of $\mathcal{U}_1$ is that it converges to the quality Q_1 if the region of interest converges to a single point. Under this condition, it can be shown that the RAST algorithm converges to the global optimum: let us assume that RAST converges to a solution $\hat{\theta}$, i.e. it produces a sequence of increasingly smaller regions $\Theta_n|_{n=K}^{\infty}$ such that $\Theta_{n+1} \subset \Theta_n$, $\hat{\theta} \in \Theta_n \, \forall n$, and $\lim_{n\to\infty} \Theta_n = \hat{\theta}$. As the upper bound $\mathcal{U}_1$ converges to the quality Q_1, it holds that:

$$Q_1(\hat{\theta}) = \lim_{n\to\infty} \mathcal{U}_1(\Theta_n).$$

At the same time, the regions Θ_n (which are iteratively picked as the first elements from q) are associated with a higher upper bound than all other regions:

$$Q_1(\hat{\theta}) = \lim_{n\to\infty} \mathcal{U}_1(\Theta_n) = \max_{\theta\in\mathbf{\Theta}} Q_1(\theta),$$

i.e. RAST converges to the globally optimal solution. In practice, search is aborted using a stopping criterion, for example if the size of the parameter region of interest falls below a certain limit.

6.4.2 Approach 2: Adding Spatial Coherence

The optimality criterion Q_1 introduced in Equation (6.7) was derived from the data likelihood, where all motion samples in the field D were assumed to be independent. In real-world scenes, however, regions of coherent motion (like the scene background and foreground objects) tend to be spatially correlated. While this fact has been ignored in the last section, in the following an extension of the criterion Q_1 is presented such that the spatial coherence of target regions is taken into account. To do so, we first introduce a notation for the fact that a sample belongs to the background. This is done using labels $L_\theta^1, ..., L_\theta^n \in \{0, 1\}$ such that $L_\theta^i = 1$ exactly if the motion sample (x_i, v_i) belongs to the background region (which again is exactly the case if $s(x_i, v_i; \theta) > 0$). Otherwise, $L_\theta^i = 0$, and we consider the sample to be an outlier. Note that these labels depend on the global motion: for some choices of θ a sample may belong to the background, while for others not.

According to the assumption of spatial coherence, sites in the motion field that are close to each other should with a high probability belong to the same region (i.e., have the same label). To express this fact, we define a neighborhood structure over the motion field sites $x_1, ..., x_n$. Sites are assumed to be arranged on a regular grid, such that *cliques* can be defined as all pairs of sites (x_i, x_j) that

are 4-connected. Letting $\mathcal{C}$ denote the set of all such cliques, the quality function Q_1 from the last section is extended with a spatial coherence term:

$$Q_2(\theta) = Q_1(\theta) + \gamma \cdot \sum_{(x_i,x_j)\in\mathcal{C}} L_\theta^i \, L_\theta^j. \tag{6.9}$$

Q_1 is the quality criterion from Equation (6.7). The parameter γ determines the weight of spatial coherence relative to the goodness-of-fit term Q_1. This extension has two major effects: first, it penalizes large foreground regions (its maximum is reached when the whole screen belongs to the background). Second, it favors solutions leading to smooth (and thus short) object boundaries. Similar terms have previously been used in other motion segmentation formulations (e.g., [Wei97]).

Optimization using RAST To optimize Q_2, an extension of the original RAST algorithm is presented that takes the spatial coherence term from Equation (6.9) into account. The basic idea, namely to perform a branch-and-bound search of parameter space, remains the same, and so does the structure of the algorithm (Table 2). The only difference is that a new bound $\mathcal{U}_2$ is defined such that $\mathcal{U}_2(\Theta) \geq Q_2(\theta) \; \forall \, \theta \in \Theta$. This bound is set to:

$$\mathcal{U}_2(\Theta) = \mathcal{U}_1(\Theta) + \gamma \cdot \mathcal{U}'(\Theta),$$

where $\mathcal{U}'(\Theta)$ denotes the upper bound for the spatial coherence term. $\mathcal{U}'$ is computed in two steps: first, for each motion sample we compute an upper bound $\mathcal{S}$ for its support $s(x_i, v_i; .)$:

$$\mathcal{S}(x_i, v_i; \Theta) \geq \max_{\theta\in\Theta} s(x_i, v_i; \theta),$$

By this, we determine for each motion sample (x_i, v_i) if the sample *could potentially* fit a motion $\theta \in \Theta$. To compute $\mathcal{S}$, we again use interval arithmetic. Then, from $\mathcal{S}$ an upper bound for the *label* L_θ^i of the corresponding motion sample is derived:

$$\mathcal{L}_\Theta^i = \begin{cases} 1 & \mathcal{S}(x_i, v_i; \theta) > 0 \\ 0 & \text{else.} \end{cases}$$

Given these upper bounds, we obtain:

$$\mathcal{U}_2(\Theta) = \mathcal{U}_1(\Theta) + \gamma \cdot \underbrace{\sum_{(x_i,x_j)\in\mathcal{C}} \mathcal{L}_\Theta^i \mathcal{L}_\Theta^j}_{\mathcal{U}'(\Theta)}$$

Note that the computation of $\mathcal{U}_2$ comes with negligible extra effort compared to $\mathcal{U}_1$, as the upper bounds $\mathcal{S}(x_1, v_1; \Theta), ..., \mathcal{S}(x_n, v_n; \Theta)$ (and with them $\mathcal{L}^1_{\Theta'}, ..., \mathcal{L}^n_{\Theta'}$) are implicitly computed as parts of $\mathcal{U}_1$ (which is simply the sum of the single-sample bounds: $\mathcal{U}_1(\Theta) = \mathcal{S}(x_1, v_1; \Theta) + .. + \mathcal{S}(x_n, v_n; \Theta)$).

6.4.3 Experiments

In the following, the proposed approach is evaluated on synthetic motion fields (where a ground truth background motion is known) and on MPEG-4 motion vector fields derived from real-world video sequences. It is illustrated that RAST optimization gives superior results compared to several local search methods in terms of motion segmentation quality, support region size, and quality of motion estimates.

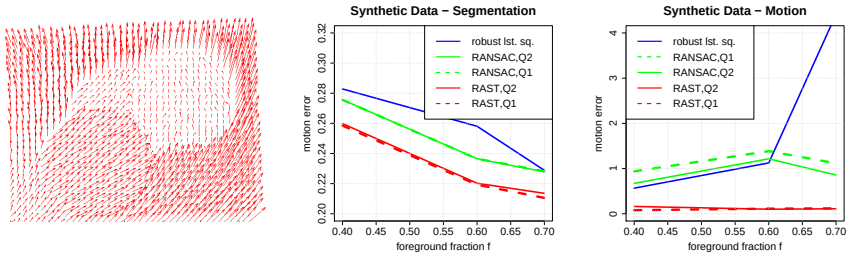
General Setup All input motion fields — synthetic or extracted from video — are defined at 16×16 macroblock positions. For video streams, motion fields are obtained using the MPEG-4 video codec XViD[8] [Tou02]. The RAST approach is compared with a variety of local search techniques. For all methods, the same outlier threshold of $\epsilon = 2.3$ (Equation (6.7)) was used, which was validated to be an appropriate choice in previous test:

1. **Least Squares**: As a baseline method, least squares parameter estimation [FP02, Section 3.1] is used, which is equivalent to assuming a Gaussian motion density instead of a truncated Gaussian one (Equation (6.6)). Outliers are not taken into account.

2. **Robust Least Squares**: Robust least squares methods alternately compute least squares estimates and discard motion samples from D that deviate further from the solution than an outlier threshold t. Starting with a high threshold $t_0 = 100$, the method iteratively (a) computes a new motion estimate based on all inliers, (b) refines the set of inliers using the threshold t_k, and (c) sets $t_{k+1} = 0.95 \cdot t_k$. This process is repeated until t_k reaches the final outlier threshold ϵ.

3. **XViD**: This is the global motion estimation component of the XViD codec. The implementation is comparable to robust least squares but uses a "greedier" outlier rejection strategy.

[8] www.xvid.org



(a) (b) (c)

Figure 6.1: (a) A synthetic motion field with three foreground blobs, each moving in a different direction. (b,c) Segmentation and motion estimation error, plotted against the size of the foreground blobs. In both cases, the RAST approach gives consistently lower error compared to local search methods ($n = 750$).

4. **RANSAC**: Random Sample Consensus (RANSAC) [FB81] is a popular fitting procedure with excellent robustness to outliers and noise. The method has also frequently been used for global motion estimation [FdW05, Sib08]. RANSAC is a stochastic algorithm: the solution is obtained by iteratively sampling a random subset $D^k \subset D$ consisting of k samples (here, $k = 2$), estimating a least squares solution θ on this subset (assuming that D^k contains no outliers), and evaluating the quality of θ on the whole motion field D. This process is repeated K times, and the best estimate is returned (after doing a least squares refinement on the set of all inliers). The probability of failure decreases with the number of iterations K, but never reaches zero — in contrast to the proposed method, optimality is not guaranteed.

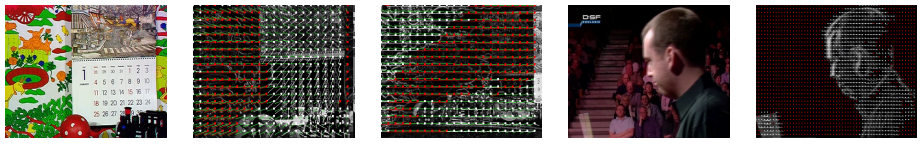
5. **RAST**: The proposed approach was tested with $\gamma = 1$, which was validated to be a appropriate choice in previous tests. The transformation space searched by RAST is set to $\sigma \in [0.9, 1.1], \alpha \in [-0.1, 0.1]$, and $(c_x, c_y) \in [-40, 40]^2$, such that it contains all reasonable motion between subsequent video frames . An accuracy of 0.1 pixels for the translation and 0.0002 for rotation and scale was found sufficient, and the convergence conditions of RAST are set accordingly. Finally, a least squares refinement on all inliers is done.

Synthetic Flow Fields In a first experiment, real-world phenomena like noise and spatial coherence are simulated on synthetic motion fields. As the example in Figure 6.1(a) illustrates, these motion fields show three blob regions moving in front of a dynamic background. The background motion is randomly drawn from $[-0.05, 0.05] \times [0.95, 1.05] \times [-10, 10]^2$. Also, three blobs are initialized with a random motion from $\{0\} \times \{1\} \times [-16, 16]^2$. These three blobs are of the same size. Together, they occupy a certain fraction of the motion field $f \in \{0.4, 0.6, 0.7\}$ (the higher f, the more difficult global motion estimation becomes). Further, isotropic Gaussian noise with standard deviation $\sigma \in \{1.0, 1.3, 1.6, 2.0, 2.3\}$ was added to each motion vector. This way, 5 motion field sequences of 10 frames each were generated for all combinations of noise levels σ and screen fractions f, obtaining a total of 750 motion fields.

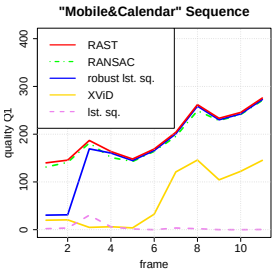
All methods are tested except XViD (which will be applied to real-world videos later) and least squares (which performed much worse than all other approaches). Results are given in Figure 6.1. In Figure 6.1(b), the average segmentation error is plotted against the fraction f occupied by the foreground. First, note that some intrinsic segmentation error results from outliers due to noise. The rate of such outliers — and thus the segmentation error — constantly drops with f. When comparing the different methods, the proposed approach gives the lowest segmentation error by a margin of $2-4\%$. This improvement is significant (paired t-test, level 99%). In Figure 6.1(c), the average error of the estimated motion (more precisely, the x-translation parameter) is plotted against f. Again, the proposed RAST approach shows a significantly better performance than other methods (paired t-test, level 99%), with a mean squared error of about 0.1 pixels. Finally, it can be observed that Q_1 and Q_2 give a similar performance for RAST. For RANSAC, the combination with spatial coherence leads to slightly better motion estimates.

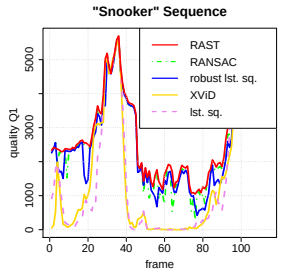
Video Sequences — Motion Estimation The proposed approach was also evaluated on MPEG-4 motion vector fields derived from test video sequences. Thereby, the quality of a motion estimate is measured in terms of the *support* Q_1. All methods were run on two test sequences (for RANSAC, 20 iterations were used). The first one called "Mobile & Calendar"[9] shows a textured background behind three foreground objects, each moving in a different direction (a subsampling at 1 fps was done). The second sequence called "Snooker" was captured from a TV sports broadcast, where a snooker player is tracked by a translating camera.

[9] http://www.m4if.org/resources.php



(a) (b) (c) (d) (e)





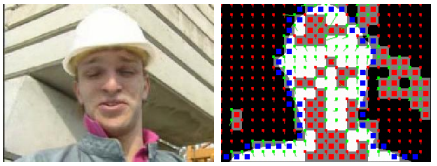
method	time(sec.)
RANSAC	55
rob. lst. sq.	153
RAST	2000
RAST (Q_2)	1300

(f) (g) (h)

Figure 6.2: (a-e): Frames from the test sequences "Mobile & Calendar" and "Snooker" with results of global motion estimation. It can be seen that RAST (b) outperforms the XViD global motion estimation (c). (f-g): motion support Q_1, plotted against the frame number. RAST gives an upper bound for all other methods, which tend to fail from time to time. (h) timing results for all methods on the "Snooker" sequence.

Results are illustrated in Figure 6.2. For either sequence, a sample frame with motion segmentation results is illustrated. Segmentation is visualized by coloring motion probes (red probes correspond to the background region, white ones to outliers). This is overlayed with a motion-compensated difference frame. For the "Mobile & Calendar" sequence, the result of RAST (Figure 6.2(b)) is compared with the XViD motion compensation (Figure 6.2(c)). It can be seen that XViD classifies parts of the background (on the upper left) as foreground and compensates them poorly. In contrast, RAST correctly identifies the background region and leads to a good estimate.

Quantitative results for both sequences are provided in Figure 6.2, where the quality Q_1 is plotted against the frame number. For both sequences, it can be seen that least squares and XViD give suboptimal results, and that robust least squares and RANSAC perform comparable to RAST but fail occasionally. Overall, RAST provides an upper bound for the performance of the other methods.



(a)

	segm. error (%)	
method	**"Curt."**	**"Forem."**
RAST	7.5	24.2
RANSAC	7.7	25.0
Rob. Lst. Sq.	7.8	25.5
XViD	11.7	33.1
Least Squares	40.6	41.4

(b)

Figure 6.3: (a) A RAST segmentation result on the "Foreman" sequence (red blocks correspond to misclassifications). (b) Segmentation error rates on the test sequences "Curtain" and "Foreman".

Figure 6.2 also provides the execution time of different methods on the "Snooker" sequence, indicating that local search procedures are significantly faster than RAST (by a factor of more than 8). Note, however, that RAST optimization could easily be parallelized (multiple threads can investigate different subregions of parameter space independently). We can also see that, when comparing the processing time of RAST with and without a spatial coherence term, that the spatial information in Q_2 leads to a speedup. Obviously, spatial coherence helps optimization to discard bad motion hypotheses early which are scattered over the frame, such that search is guided into promising regions of transformation space more quickly. This insight might also be interesting in the geometric matching domain where RAST was developed originally.

Video Sequences — Motion Segmentation Finally, the proposed approach is evaluated with respect to motion-based segmentation. Tests are run on two video sequences with known ground truth segmentations: first, the self-generated "Curtain" sequence of 700 frames with two hands moving in front of a green curtain (ground truth segmentations were extracted using a histogram-based skin color model [JR02]). Second, a subsampled version of the well-known "Foreman" sequence (80 frames) that comes with manually generated ground truth masks.

Numerical results in terms of segmentation error are also given in Figure 6.3 (the extension with spatial coherence was used). They confirm observations made in previous tests: the proposed approach provides an optimal performance and gives moderate improvements over other test methods. Segmentation errors occur for frames where the object stands still or due to a complicated scene structure for which the 4D similarity transform model (Equation (6.4)) is not adequate. The most severe problem, however, are errors in the previous motion estimation step.

6.4.4 Discussion

In this section, the estimation of a global parametric motion from an optical flow field has been addressed. A novel approach based on the RAST algorithm has been proposed, which performs a full, adaptive search of transformation space and thus — in contrast to local search procedures — gives an optimal result.

It should be kept in mind that the proposed approach is limited in two ways: first, it is restricted to low-dimensional motion parametrizations (for the experiments in this section, a 4D similarity transform [Equation (6.4)] was used), and a search of high-dimensional parameter spaces is cost-intensive. Second, while the proposed approach was demonstrated to achieve a better accuracy, results also show that local search techniques are significantly faster and thus remain an appealing alternative in real-world video processing scenarios.

Yet, RAST is the method of choice if a high accuracy of motion estimation is desired and time constraints are weak. Also, the proposed approach can serve as a generator of ground truth for evaluating other motion estimation methods, as it provides the best motion estimate achievable for a given motion model and video sequence.

Finally, as our segmentation results demonstrate, a general weakness of the feature-based framework lies in the fact that errors in the motion estimation step cannot be overcome in the motion interpretation step addressed here. Though this problem can be addressed to some extent using robust features and matching techniques [TZ00], it also motivates *direct* methods targeted at a joint motion estimation and segmentation. Such an approach will be investigated in the following section.

6.5 Segmentation by Combining Motion Information with Color Models

In this section, a second model for motion segmentation is presented that follows a *direct* strategy [IA00], i.e. it couples motion estimation and scene segmentation in a joint process directly operating on the pixel intensity. The benefit of this approach lies in a higher robustness to errors and uncertainty of motion estimation. In contrast to the last section, the focus is not on optimization (for which we will refer to graph cut [BK04] as a standard technique). Instead, a combination of motion segmentation with *color* information is proposed. Both motion and color can be useful clues for scene segmentation, but remain fairly limited if employed

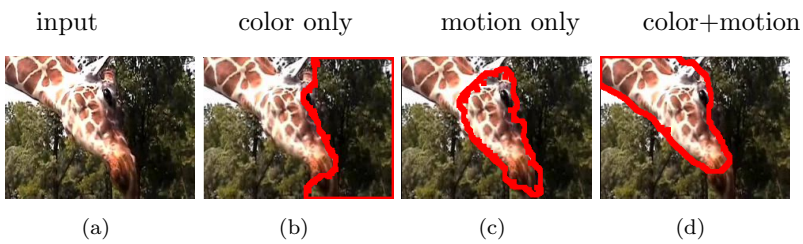


Figure 6.4: An illustration of combining color and motion segmentation. While color (b) and motion (c) information alone lead to incorrect segmentations, a simple combination of both as described in this section (d) segments the object from the background correctly (picture from YouTube).

individually. On the one hand, color can help to group images into regions of coherent appearance [SM00], but the resulting segments cannot be assumed to correspond to meaningful objects in general [MFTM01]. On the other hand, motion information provides that regions can often be associated with objects, but — as previously outlined — inaccuracies occur due to illumination changes, specular highlights, motion discontinuities, transparency, etc. Correspondingly, a segmentation based on motion alone remains a challenging problem as well. The fact that motion and color clues individually do not provide reliable segmentations is illustrated in Figure 6.4, where sample results for a video scene showing a moving giraffe are illustrated. A color-based approach partitions the scene into a dark and bright area but does not give a segmentation of the object. Motion segments only the head but misses the neck.

This raises the question whether segmentation can be improved by combining both sources of information. In this section, a simple and efficient extension of motion segmentation with color models is presented that partitions video frames into regions of coherent motion *and* color (in Figure 6.4(d), this approach is illustrated to give a good segmentation of the object). The method decomposes a dynamic scene into a foreground and background layer. For each layer, statistical models are used to describe the motion and color within. Motion is modeled with the approach by Schoenemann and Cremers [SC06]. For color, several choices like histograms and Gaussian mixtures are tested. Cost terms are formulated for assigning each pixel to the foreground or background, and for assigning neighbor pixels to different layers. Both kinds of terms serve as edge weights in a cost graph, and a segmentation is obtained using a graph cut algorithm.

6.5.1 Related Work

The idea of integrating multiple clues for object segmentation in video has been followed before by a variety of methods. These are usually subsumed under the term *spatio-temporal* segmentation, as they make use of both spatial information drawn from the frame of interest and temporal information in form of differences to previous and subsequent frames.

Frequently, spatio-temporal methods use heuristic approaches to integrate motion with other clues, such that texture, intensity, or color are used in additional preprocessing or postprocessing steps. One very simple approach is to adapt motion boundaries to image edges after motion segmentation. For example, Mech and Wollborn [MW98] apply a change detection to motion-compensated frames and then align motion boundaries with image edges detected with a Sobel operator. Piroddi et al. [PV02] obtain separate segmentations from color, motion, and texture, and fuse them using morphology and heuristic rules. Altunbasak et al. [AET98] also follow the idea that motion boundaries usually coincide with image edges. Therefore, previous to motion analysis a segmentation based on color is performed. The resulting oversegmented regions are then tracked and finally grouped according to their individual parametric motion. This approach offers benefits in terms of processing speed, as motion segmentation is performed on the level of coarse regions. Also, motion boundaries coincide inherently with intensity boundaries in the image. A similar idea is followed by Choi et al. [CLK97], who estimate a few initial regions of coherent motion *and* color as reliable initial markers. Then, a grouping is applied in which spatial regions obtained by a watershed segmentation are assigned to these markers.

Other approaches closer to the one presented in this section are targeted at a joint application of motion and texture. Typically, segmentation is cast as an energy minimization problem, whereas cost terms are defined for assigning pixels to each scene layer. In this framework, color and intensity clues can be integrated by defining cost terms not only based on motion but also on spatial information. A variety of methods follow this approach: Black [Bla92] presents an early system based on Markov Random Fields. Cost terms include both intensity and motion, and spatial coherence of segmentation is enforced by penalizing local constellations of image boundaries. Kahn and Shah [KS01] present another system, where combined cost terms are derived from the assumption of independent motion and color clues. This idea is adopted, but smoothness terms are added for the motion boundary. Also, while Kahn and Shah use pre-computed optical flow, the approach proposed here couples motion estimation and segmentation in a joint process.

More recently, graph cut algorithms [BK04] have become a popular approach for segmentation. Thereby, energy functions are optimized that combine data fit terms with smoothness terms penalizing an assignment of neighbor pixels to different layers. Both kinds of terms serve as edge weights in a cost graph, such that segmentation effectively becomes a two-way graph cut problem (for which an optimal solution can be obtained in polynomial time [BK04]). Several approaches have been presented based on this idea. Li et al. [LSS05] suggest a semi-automatic method, where manual segmentations are provided for certain keyframes. These segmentations are propagated over time using graph cut, obtaining segmentations of other frames. Galun et al. [GAB05] present another system in which graph cut segmentation is used in a multi-scale approach. Based on probabilistically motivated similarity measures, image regions are grouped together if they fit a common motion model. Initially, simple translational models are used, which are later replaced with higher-dimensional, more complex alternatives (affine flow and epipolar geometry). This allows the grouping of increasingly heterogeneous and larger motion segments. Again, intensity is used as part of a region-level similarity measure. Another approach by Wang et al. [WXZG07] uses initial motion estimates from edge and corner pixels to obtain reliable motion parameters. The remaining pixels are then labeled using a graph cut approach, in which color differences are integrated together with a spatial coherence term.

Though the approach presented in this section strongly resembles such graph-based methods, there are two differences. First, a joint parameter estimation and segmentation is performed, while the aforementioned methods estimate motion in a preprocessing step [WXZG07], adapt motion and color parameters from previous frames [Bla92], or neglect motion altogether [LSS05]. Second, experimental results in previous work do usually not provide a comparison with motion-only baselines. In contrast to this, the approach presented in this section will be evaluated on different kinds of video data, and improvements will be validated compared to using motion clues only.

Finally, another category of approaches called *sprite-based* methods [JF01, KTZ05a, KTZ08] follows a different way of combining motion and appearance. A model of the video scene is maintained consisting of image fragments (or *sprites*, respectively). A generative process for video frames is defined by transforming and morphing these sprites and overlaying them in the image plane. Sprites, motion, and the order of layers in the video are learned in a joint optimization process, where each component is alternately optimized while fixing the others. This approach offers an elegant handling of motion blur and occlusion [KTZ08]. Compared to such methods, the work proposed in this chapter can be viewed as a light-weight

approach, where pixel-exact models of scene layers are replaced with much simpler global color models. As pointed out by Jojic et al., such light-weight methods are less prone to problems caused by local minima [JWZ06] and are significantly faster. While the estimation of a sprite-based approach has been reported to require several minutes per frame [KTZ08], the proposed method runs in near-realtime. In fact, it has already been proposed to combine both kinds of models such that a selection between complex pixel-accurate sprite models and simple motion models is made automatically during optimization [JWZ06].

6.5.2 Approach

We follow the notation introduced in Section 6.2, denoting pixel intensity with $I(x,t)$ and RGB values with $I_R(x,t)$,$I_G(x,t)$,$I_B(x,t)$ defined over positions x in frames t. Motion is derived from the spatio-temporal derivatives $\nabla I(x,t) = (\,\partial I(x,t)/\partial x_1,\, \partial I(x,t)/\partial x_2\,)$ and $I_t(x,t)$. These measurements are combined in a joint vector of image data $\mathcal{I}(x,t) = (\nabla I(x,t), I_t(x,t), I_R(x,t), I_G(x,t), I_B(x,t))$. As only the gray level parts of this vector are used for motion estimation, we divide $\mathcal{I}(x,t)$ into a motion part $\mathcal{I}_v(x,t) = (\nabla I(x,t), I_t(x,t))$ and a color part $\mathcal{I}_c(x,t) = (I_R(x,t), I_G(x,t), I_B(x,t))$.

Let us first focus on the segmentation of a single frame t. In this case, we can simplify notation by dropping the time component t and abbreviating $I(x,t)$ with $I(x)$, $I_t(x,t)$ with $I_t(x)$, etc. The goal of segmentation is to estimate a bi-level mask m such that $m(x) = 1$ exactly if the pixel x in frame t belongs to the foreground (and $m(x) = 0$ otherwise). Also, a parameter vector θ is inferred that describes the motion and appearance in the foreground and the background.

We formulate segmentation as an energy minimization problem [BF93, SC06]:

$$\begin{aligned}
\hat{m}, \hat{\theta} &= \arg\min_{m,\theta} \quad E_1(m, \theta; \mathcal{I}) \\
&= \arg\min_{m,\theta} \quad \underbrace{\alpha \cdot \sum_x -\log p(\mathcal{I}_c(x) \,|\, m, \theta_c)}_{color\ cost} \\
&\qquad + \underbrace{(1-\alpha) \sum_x -\log p(\mathcal{I}_v(x) \,|\, m, \theta_v)}_{motion\ cost} \\
&\qquad + \underbrace{\sum_{(x,y)\in\mathcal{C}, m(x)\neq m(y)} \beta}_{smoothness\ cost},
\end{aligned} \tag{6.10}$$

where $\mathcal{C}$ is the set of all mutually 4-connected pairs of neighbor pixels. E_1 consists of three terms: the first two "data fit" terms regulate the fit of pixels to scene foreground and background. For every pixel, a *color likelihood* $p(\mathcal{I}_c(x) \,|\, m, \theta_c)$ and a *motion likelihood* $p(\mathcal{I}_v(x) \,|\, m, \theta_v)$ are formulated, each with distinctive motion and color parameters, θ_c and θ_v. The weight $\alpha \in [0, 1]$ balances the influence of color and motion information. The last term enforces the boundary between regions to be smooth, with $\beta > 0$ weighting the importance of the prior relative to the previously defined likelihood terms. Particularly, our focus is on the additional color term, which will be evaluated experimentally later. Note that if $\alpha = 0$, this color information is neglected and the model boils down to a purely motion-based approach [SC06], which will serve as a baseline in later experiments. In the following, the motion and color cost terms are addressed in more detail, i.e. statistical models for the likelihoods $p(\mathcal{I}_v(x)|m, \theta_v)$ and $p(\mathcal{I}_c(x)|m, \theta_c)$ are discussed.

Motion Information The model for the motion likelihood $p(\mathcal{I}_v(x)|m, \theta_v)$ is adopted from an approach by Schoenemann and Cremers [SC06]. It is based on the assumption of separate parametric motions within foreground and background regions. Like in [SC06], affine motion models (Equation (6.2)) and constant flow ($v_\theta(x) = (c_1, c_2)$) will be tested.

Two motions are defined, one for the background layer and one for the foreground. The associated parameters are denoted with θ_v^b and θ_v^f. For each pixel position x, the model predicts the associated motion vectors assuming that the pixel belongs to the foreground, $v_f(x)$, and to the background, $v_b(x)$. The quality of these motion predictions is measured using optical flow error:

$$\begin{aligned} e_f(x) &= \nabla I(x) \cdot v_f(x) + I_t(x), \\ e_b(x) &= \nabla I(x) \cdot v_b(x) + I_t(x). \end{aligned}$$

If $v_f(x)/v_b(x)$ is an accurate motion prediction, the associated flow error is zero according to the optical flow equation (6.1). In practice, however, inaccuracies occur due to phenomena such as camera noise, illumination changes, and specular highlights. Therefore, error is assumed to be normally distributed with mean 0 and variances $\sigma_b \cdot ||\nabla I(x)||^2$ and $\sigma_f \cdot ||\nabla I(x)||^2$. This leads to the following likelihoods for foreground and background:

$$\begin{aligned} p_f(\mathcal{I}_v(x) \,|\, \theta_v^f) &= \mathcal{N}(e_f(x)\,;\, 0, \sigma_f^2 ||\nabla I(x)||^2) \\ p_b(\mathcal{I}_v(x) \,|\, \theta_v^b) &= \mathcal{N}(e_b(x)\,;\, 0, \sigma_b^2 ||\nabla I(x)||^2), \end{aligned}$$

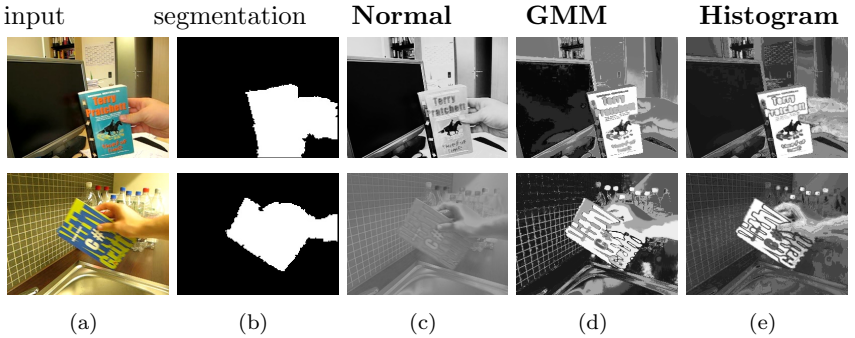


Figure 6.5: Given input images (a) and segmentations (b), color models for foreground and background are trained and used to estimate pixel score maps (c,d,e). Bright regions have a high probability of belonging to the foreground according to the color model. It can be seen that Normal densities (c) perform poorly, while Gaussian mixtures (d) and color histograms (e) discriminate foreground and background better.

and the overall motion model with motion parameters $\theta_v = (\theta_v^b, \sigma_b^2, \theta_v^f, \sigma_f^2)$ becomes:

$$p(\mathcal{I}_v(x) \,|\, m, \theta_v) = m(x) \cdot p_f(\mathcal{I}_v(x) \,|\, \theta_v^f) + (1 - m(x)) \cdot p_b(\mathcal{I}_v(x) \,|\, \theta_v^b). \tag{6.11}$$

Color Information Similar as for motion, color in the foreground and background is modeled using parametric distributions, now over color features $\mathcal{I}_c(x)$. To model these distributions $p_f(\mathcal{I}_c(x)|\theta_c^f)$ and $p_b(\mathcal{I}_c(x)|\theta_c^b)$, different choices are possible. One option are normal densities (where the parameters $\theta_c^{b/f}$ become mean values and variances in RGB color space). Others are Gaussian mixture models (GMMs) or color histograms (where $\theta_c^{f/g}$ are associated with histogram bins). Several choices are illustrated in Figure 6.5. Given a segmented input image, different color models for foreground and background are trained, and a posterior for each pixel to belong to the foreground is inferred:

$$P(m(x) = 1 \,|\, \mathcal{I}_c(x)) = \frac{p_f(\mathcal{I}_c(x) \,|\, \theta_c^f)}{p_f(\mathcal{I}_c(x) \,|\, \theta_c^f) + p_b(\mathcal{I}_c(x) \,|\, \theta_c^b)}$$

This score is plotted for different color models, namely normal densities with diagonal covariance, Gaussian Mixture models (GMMs), and color histograms. Bright values correspond to high probabilities of foreground, illustrating how well a color

Algorithm 3 Dynamic scene segmentation

initialize m (for example with the segmentation result from the previous frame).
repeat
  estimate θ_c^f and θ_c^b (for example, by computing color histograms).
  estimate θ_v^f and θ_v^b [SC06, Section 4.3].
  re-estimate m by fixing θ, constructing a cost graph, and applying graph cut optimization.
until m does not change
return m

model discriminates between foreground and background (a perfect scoring would be the segmentation mask itself). It can be seen that normal densities give a poor discrimination, while Gaussian mixtures and color histograms separate the foreground more reliably. These observations will be confirmed in terms of segmentation error later.

Optimization To estimate the segmentation mask m and parameters θ, the energy E_1 is minimized given image data $\mathcal{I}$ and weights α, β. Optimization is carried out in an iterative scheme similar to the one in [SC06], where θ and m are estimated alternately. For an outline and illustration, please refer to Algorithm 3 and Figure 6.6.

An efficient algorithm described by Boykov and Kolmogorov [BK04] is used for solving the graph cut problem. Also, the motion model can be estimated efficiently from statistics of flow error. At a resolution of 160 × 120 pixels, an untuned prototype runs at 3 fps on a 2.4 GHz machine.

Extensions: Shape and Contrast The model from Equation (6.10) can be extended using two observations. The first one is that motion boundaries tend to coincide with image edges, which motivates the use of a **contrast term** similar to the one by Kumar et al. [KTZ05a]:

$$E_2(m,\theta;\mathcal{I}) = E_1(m,\theta;\mathcal{I}) - \beta \cdot \eta \sum_{(x,y)\in\mathcal{C}, m(x)\neq m(y)} \left[1 - exp(-\frac{(I(x)-I(y))^2}{2\sigma^2})\right] \qquad (6.12)$$

i.e., the smoothness cost from Equation (6.10) is reduced by a factor $\eta \in [0,1]$ depending on the gray value pixel difference (like Kumar et al. [KTZ05a], we estimate σ^2 as two times the mean squared pixel difference).

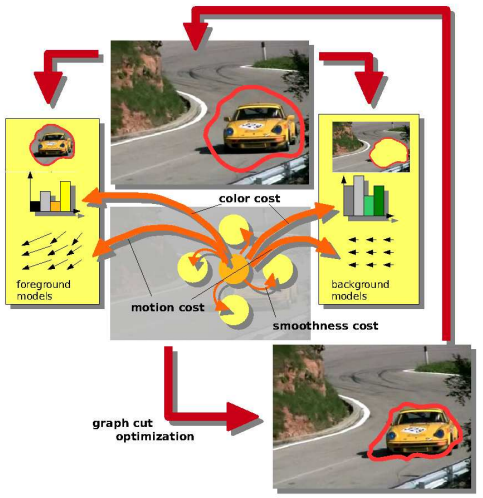


Figure 6.6: An illustration of video segmentation as an iterative optimization over color and motion. Based on the current segmentation (top), color and motion models for foreground and background are estimated (left and right). All pixels in the image are then fitted to these models, obtaining color and motion cost. These are associated with edges in a cost graph, and graph cut is used to infer a new segmentation. This process is repeated until convergence.

The second observation is that objects move smoothly between successive frames. Therefore, while the previous formulations E_1 and E_2 focused only on a certain point in time, the problem is now extended to video sequences. A sequence of frames $\mathcal{I}(.,1),...,\mathcal{I}(.,T)$ is assumed to be given, and segmentation masks $m(.,1),...,m(.,T)$ are to be inferred. Correspondingly, different parameters for different frames $\theta_1,...,\theta_T$ are estimated. The previous formulation is extended such that the mask $m(.,t)$ in frame t is constrained to be similar to the one from the previous time step, $m(.,t-1)$, using an additional **shape consistency** term [KS01, WXZG07]:

$$E_3(\, m(.,t), \theta^t \,;\, \mathcal{I}(.,t), m(.,t-1)\,) = E_2(m(.,t), \theta^t; \mathcal{I}(.,t)) + \sum_{x:m(x,t)\neq m(x,t-1)} \gamma \qquad (6.13)$$

where the parameter γ regulates the influence of shape consistency. It should be noted that this extension helps in cases where motion is not discriminative. For example, if the object stands still for a moment, the proposed approach relies on color and shape consistency clues and can still give a correct segmentation.

Finally, it should also be kept in mind that the extensions in E_2 (contrast) and E_3 (shape consistency) do not change the structure of the optimization problem. The additional terms simply turn into modifications of edge costs in the graph.

6.5.3 Experiments

The proposed approach is evaluated in the following experiments, where the influence of color information compared to motion-only segmentation is studied. It is a common problem with segmentation that ground truth information is difficult to obtain at a large scale — in fact, many publications only provide limited results on a few test sequences. To give a more thorough evaluation, two experiments are presented in the following. Experiment 1 is conducted in a constrained setup with a static camera, for which *background subtraction* techniques can give segmentations of sufficient accuracy to serve as ground truth. After this, a smaller-scale experiment for dynamic scenes using manually generated ground truth is presented.

Static Scenes In case of a static background and fixed camera, an almost perfect segmentation can be achieved using robust background subtraction techniques [SZTS06]. This provides a simple way for generating ground truth automatically at a larger scale compared to a tedious manual segmentation. A dataset of 24 video clips of ca. 3 seconds length was used, in which several objects were presented to a camera in front of varying static backgrounds. 507 frames were sampled (frames without foreground objects in them were ignored). Ground truth segmentations were obtained automatically using a self-implemented background subtraction (as this serves only as a source of ground truth, a closer description is omitted here). The resulting segmentations were briefly checked to be correct. A sample result in Figure 6.7(b) indicates that an accurate segmentation is achieved. It should be kept in mind, however, that this approach is restricted to static scenes and is therefore significantly limited compared to the more general setup of motion segmentation studied in this chapter.

Previous to motion segmentation, frames were scaled to a resolution of 160×120 pixels. An affine motion model was used with parameters $\beta = 6, \gamma = 2.5$, and $\eta = 0.5$ (this setup was obtained by a grid search optimization of segmentation error). As the goal of the experiment is to evaluate the influence of color information on the system, the color weight α was varied. If $\alpha = 0$, the system uses only motion information. With α, the influence of color on the segmentation increases.

A sample result is illustrated in Figure 6.7. In this scene, motion alone is not sufficient to segment the object from the background (possible explanations are

(a) (b) (c) (d)

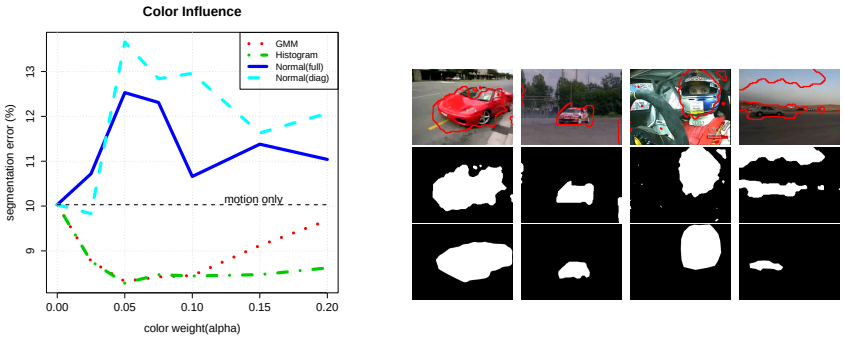
Figure 6.7: Illustration of the "static scenes" experiment: (a) an input image. (b) Ground truth obtained using a background subtraction approach. (c) Segmentation result using motion only. (d) Segmentation result when combining color and motion information.

sudden illumination changes and a motion pattern that does not properly fit the parametrization). If combined with color information, however, the object can be segmented almost perfectly.

Quantitative results are given in Figure 6.8(a), where the segmentation error is plotted against the color weight α. Four color models were tested: a Gaussian mixture model (12 components, fitted using Expectation Maximization [DLR77]), a color histogram model (10^3 bins), and as baselines Normal densities with full and diagonal covariance matrices. It can be seen that by choosing a moderate color influence ($\alpha \approx 0.05$), segmentation error can be improved from 10% to 8% compared to a purely motion-based segmentation. According to a t-test (level 99%) this improvement is significant. When comparing different color models, we see that a model of a certain complexity is advantageous - while both normal densities fail, the mixture model and the histogram lead to comparable improvements.

Dynamic Scenes A second experiment is conducted for dynamic scenes in which the background is allowed to move. As in these cases no automatic generation of ground truth is possible, 30 pairs of frames were segmented manually. The data was sampled from videos downloaded from the video portal *revver.com* and shows cars, faces, and animals in motion.

In this experiment, larger images were used (240×180 pixels) and also smoothed with a Gaussian filter previously to segmentation. We tested the system with a constant motion model and a histogram color model. Compared to the static case, results show higher error rates (which can be explained by the fact that the segmentation problem is more difficult for dynamic scenes), but also a reduction from 13.5% ($\alpha = 0$) to 12.7% ($\alpha = 0.3$) by using color information. Some sample



(a) (b)

Figure 6.8: (a) Quantitative segmentation results for static scenes. Segmentation error is plotted against the color weight α ($n = 507$). For GMMs and color histogram models, segmentation error can be reduced from 10% to 8%. (b) Sample results for dynamic scenes. Top row: input. Center: results. Bottom row: ground truth (manually acquired). Pictures from revver.com.

segmentations are illustrated in Figure 6.8(b). Overall, this demonstrates that the proposed way of integrating color clues improves segmentation in both static and dynamic scenes.

6.6 An Object Recognition Framework using Motion Segmentation

In Sections 6.4 and 6.5, the challenge of segmenting video scenes into layers of coherent motion has been addressed, and it has been demonstrated that segmentations of physical objects from the scene background can be obtained. The work presented in the following is targeted at using this information for an improved recognition of objects. Thereby, a patch-based approach is followed for recognition, which is a popular approach and has been pointed out to achieve high robustness with respect to pose variation, illumination changes, clutter, and intra-class variation [Low04, PCI+07, PHSZ07, Rot08] (for a discussion, please refer to Section 6.3.2).

In the following, a simple combination of a patch-based object recognition with motion segmentation will be presented. Motion segmentation is used as a *filter*

such that only patches from the object region are used for recognition. The key question studied in this section is whether adding such segmentation information improves the robustness of a state-of-the-art object recognition. At a first glance, this might not seem very surprising. However, there is also evidence supporting the assumption that motion segmentation does not necessarily lead to an improved recognition of objects:

- Motion-based segmentation is inaccurate and error-prone, as dealing with phenomena such as illumination changes, specularity, absence of texture, and motion discontinuities is difficult. In cases of failure, such a segmentation must be considered harmful and not beneficial for recognition.
- Object recognition by itself can achieve a certain robustness. Particularly for patch-based recognition methods, it has been demonstrated that objects can successfully be discovered in cluttered scenes [FTG06, Low04, SZ06]. This is achieved by a robust matching of local features, during which unreliable correspondences are discarded based on their appearance. To some extent, this achieves a filtering of clutter similar to a segmentation.
- Finally, there may be situations in which scene background can be helpful, particularly if it is correlated with objects of interest. For example, instances of the category "airplane" tend to appear in front of sky, such that the presence of sky in the image might be a discriminative clue for a detector.

Due to these reasons, it is not obvious a priori whether motion segmentation does indeed help to improve recognition. Therefore, the proposed framework is evaluated in two experiments, one concerned with the recognition of specific objects and one with the detection of object categories in video databases.

6.6.1 Approach

In this section, an approach for the recognition of moving objects in video is presented that combines a patch-based recognition strategy with motion-based segmentation. The approach describes video frames as a collection of local image parts (or *patches*). Each patch $p = (x, f)$ comes with a position in the image x (here, the patch center) and a feature vector f describing its appearance. The core of recognition is a matching between patches in the input image and in a knowledge base of labeled object views. If a sufficient number of correspondences for a certain object is found, this object is returned as a recognition result. This

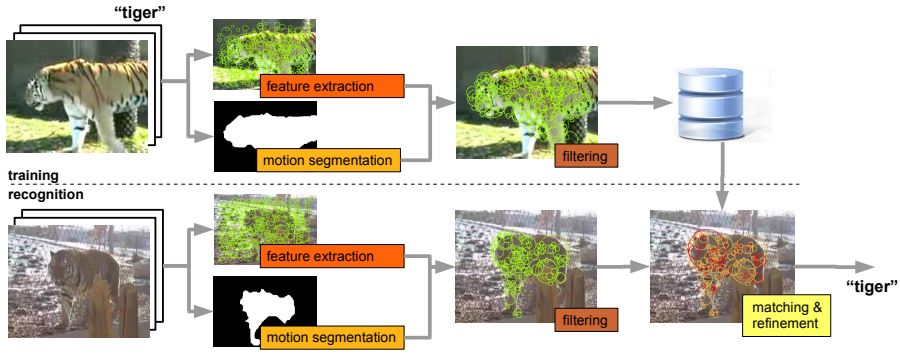


Figure 6.9: A recognition system using motion segmentation. In both training and testing, features are extracted and filtered according to a motion-based scene segmentation. A matching determines correspondences between training and test views (indicated by patches of different color). The resulting matches are used to infer the presence of an object (pictures from YouTube).

general approach can be considered a state-of-the-art technique, and variations of it are implemented in research prototypes and commercial systems [JDS08a, Low04, MPDB+06, PCI+07, SZ06].

The framework proposed in the following extends this by using motion segmentation as a filter, such that only patches situated within the foreground region are used. The whole approach is illustrated in Figure 6.9, and a detailed listing is given in the following. The processing steps in training and testing are widely the same: features are extracted, filtered by motion-based segmentation, and stored in a database (in training) or matched with object patches (in recognition). The single processing steps are outlined in the following in more detail.

Feature Extraction As already mentioned, a robust patch-based feature representation is chosen. Regions of interest are detected in the image (which can correspond to corners, blobs, or other characteristic features). The appearance of these regions is described by *local descriptors* (for a discussion of methods, please refer to Section 6.3.2). In the following, SURF features [BTvG06] will be used.

Segmentation For segmentation purposes, the fact is employed that objects correspond to regions of coherent motion in video streams. The approach from

Algorithm 4 Recognition using motion segmentation: training

given: video frames $I_1, ..., I_T$ with object label $l_1, ..., l_T$.
initialize the database: $D = \emptyset$
for all training images I_t: **do**
feature extraction: extract patches $p_1^t, ..., p_n^t$
motion segmentation: segment I_t, obtaining a mask m_t such that $m_t(x) = 1$ exactly if x belongs to the foreground.
filtering: filter out patches from the background: $D_t = \{(p_j^t, l_t) \,|\, m_t(x_j^t) = 1\}$
storage: store patches in the database: $D = D \cup D_t$
end for

Section 6.5 is used, which is based on a combination of motion and color information with graph cut optimization. The method will be referred to as *dynamic scene segmentation* in the following.

Note that the approach decomposes videos into at most *two* layers, which we assume to correspond to a moving foreground object and the scene background. It may not be clear a priori which of the segmented regions corresponds to the object and which one to the background. When recognizing objects in a video, this problem can be overcome by simply testing all segmented regions for object presence. For system training, however, the problem is more difficult, as we must simultaneously learn the object model and estimate object regions. In this chapter, a simple heuristic will be used that works well for most practical situations: as objects usually appear in the center of the frame and are surrounded by background, the region with most pixels on the image border is chosen to be the background.

Finally, it should be noted that — if segmentation is dropped, which corresponds to setting $m(x) = 1 \; \forall x$ — no features are filtered, and the approach boils down to a plain image-based recognition [Low04, SZ06]. This will serve as a baseline in later experiments.

Filtering From feature extraction, we obtain a set of patches $D = \{p_1^t, ..., p_n^t\}$ for each image I^t, with the center of patch p_i^t at location x_i^t. Further, segmentation provides a mask m^t. This information is combined in a filtering, obtaining a set of patches from the foreground region (as shown in Figure 6.9):

$$D_t = \{p_i^t \in D \,|\, m^t(x_i^t) = 1\}$$

Algorithm 5 Recognition using motion segmentation: testing

given: a video sequence $V = I_1, ..., I_T$, a database of labeled patches D, and object labels $1, ..., C$

initialize the set of matches: $M = \emptyset$

for all test frames I_t: **do**

feature extraction: extract patches $p'_1, ..., p'_m$

motion segmentation: segment I_t, obtaining a mask m_t with $m_t(x) = 1$ exactly if x belongs to the foreground.

filtering: filter out patches from the background: $D_t = \{p'_j | m_t(x'_j) = 1\}$

matching: find correspondences between D_t and the database D using a feature similarity relation '$\sim$': $M = M \cup \{(p'_i, p_j, l) \,|\, f'_i \sim f_j, f'_i \in D_t, (p_j, l) \in D\}$

end for

refinement: refine matches M based on the spatial location of patches.

scoring: compute object scores $P(c|V)$ from M

Matching The key component of recognition is a matching of features in the input image with features from a database of object views. Patches in the database are denoted with $(p_i, l_i)_{i=1}^n$, where l_i is an object label and $p_i = (x_i, f_i)$ a patch (for matching, only the appearance f_i will be of interest). Correspondingly, we assume patches $p'_1, ..., p'_m$ from a test frame to be given (which were optionally filtered using motion-based segmentation). Finally, labels $1, ..., C$ associated with C objects are assumed to appear in the database. Between training and test patches, a similarity search is performed, obtaining a set of correspondences $M = \{p_i \sim p'_j\}$. From these correspondences, object scores $P(c|V)_{c=1}^C$ are computed. We follow two general strategies, a *full patch search* and a faster approximation based on a discretization of patches to *visual words* (for more information, please refer to [JDS08b]):

Full Patch Search: For each patch in the test image $p'_i = (x'_i, f'_i)$, the nearest neighbor $nn(f'_i) = \arg\min_{j=1,...,n} ||f'_i - f_j||_2$ in the object database is found, and both are assumed to match $(p'_i \sim p_j)$ exactly if $j = nn(f'_i)$. This correspondence induces a *vote* for the object associated with $nn(f'_i)$. We aggregate votes from all patches inside the test video V in a sum rule fusion. This can be interpreted as a probabilistic object score:

$$P(c|V) = \frac{1}{m} \sum_{i=1}^{m} P(c|x'_i) = \frac{1}{m} \sum_{i=1}^{m} \delta(c, l_{nn(f'_i)}).$$

One problem is that patches in cluttered regions cast error-prone votes, which again lead to incorrect object scores. To improve the robustness of recognition with respect to clutter, Lowe [Low04] suggested to remove inconfident votes based on the *nearest neighbor ratio*: besides the nearest neighbor $nn(f'_i)$, we also find a second neighbor as the closest patch from a different object category:

$$nn_2(f'_i) = \underset{j=1,...,n:l_j \neq l_{nn(f'_i)}}{\arg\min} ||f'_i - f_j||_2,$$

and votes are accepted only if the ratio of the distances to these neighbors is smaller than a threshold $\lambda \in]0, 1]$:

$$P(c|V) \propto \sum_{i: \frac{||f'_i - f_{nn(f'_i)}||_2}{||f'_i - f_{nn_2(f'_i)}||_2} \leq \lambda} \delta(c, l_{nn(f'_i)}), \tag{6.14}$$

Obviously, if $\lambda = 1$, no filtering takes place, and the further λ is decreased the more matches are rejected. This filtering offers a certain robustness to clutter: matches with the background region are usually not distinctive for a certain object, i.e. their nearest neighbor ratio is high. Correspondingly, clutter is filtered during the matching process, which offers an alternative to motion segmentation. In later experiments, this strategy will be evaluated.

Visual Words: Another problem with full patch search is that it requires a nearest neighbor evaluation on patch basis, which can get time-consuming if the number of objects to be learned (and with it the number of patches in the database) is high. A faster recognition can be achieved by discretizing patches into a lower number of clusters (or *visual words*) $q_1, ..., q_K$, which are estimated using K-means (visual words have already been used in previous chapters, see Section 3.5). Each patch is mapped to its closest visual word, which induces a *quantizer* of patch descriptors f:

$$q(f) = \underset{k=1,...,K}{\arg\min} ||f - q_k||_2$$

It is counted how often each visual word appears in the video $V = I_1, ..., I_T$, and this information is stored in a histogram $h_1(V), ..., h_K(V)$ (the so-called *bag-of-visual-words* feature):

$$h_k(V) \propto \sum_{i=1}^{m} \delta\left(q(f'_i), k\right)$$

Similar histograms of feature counts are stored for all objects in the database:

$$h_k(c) \propto \sum_{(p_i, l_i) \in D: l_i = c} \delta\left(q(f_i), k\right). \tag{6.15}$$

Figure 6.10: Frames from the dataset used in Experiment 1. One video was captured for each combination of these 12 books and 12 indoor locations. The dataset shows significant clutter and changes of pose and illumination.

All histograms are normalized to sum one. Matching is simplified by assuming a correspondence to occur exactly if two patches belong to the same visual word. This way, matching with the full patch dataset is replaced by a matching with the cluster centers, and voting is conducted on the basis of visual words. The patch score turns out to be:

$$P(c|V) \propto \sum_{k=1}^{K} h_k(V) \cdot h_k(c) \tag{6.16}$$

Refinement Two ways of refining the correspondences obtained in feature matching have already been outlined, the first one based on the nearest neighbor ratio (Equation (6.14)), the second one (which is the key contribution of this chapter) on a motion-based segmentation of the frame. Object recognition systems sometimes add another refinement of correspondences based on patch *position*. Here, the key idea is that object features come with a restricted spatial constellation, while this does not hold for false positive matches. This makes it possible to employ additional filters on the match set (an overview has been given in Section 6.3.2).

In the proposed framework, an approach suggested by Lowe [Low04] is followed: given a set of correspondences with positions $x_1, ..., x_n$ and $x'_1, ..., x'_n$, it is assumed that the former can be mapped to the latter using an affine transformation. The parameters of this transformation are estimated from the noisy match set using a two-stage approach: a voting in Hough space is conducted to get candidate param-

(a) (b) (c)

Figure 6.11: Results of motion segmentation. (a) Input Frames. (b) Background subtraction results, which are close to perfect. (c) Results of dynamic scene segmentation from Section 6.5, which does not employ the assumption of a static background.

eters, and a refinement with robust least squares leads to a final transformation. Finally, matches are identified that fit the estimated affine transformation within an error range of ϵ. All other matches are discarded.

6.6.2 Experiment 1: Object Recognition

In the following, the proposed framework of combining object recognition with motion segmentation will be evaluated in two experiments. The first one is concerned with the recognition of objects presented to a camera. This setup is not only of interest from a practitioner's view (where objects of interest might be commercial products like books or CDs[10], items that a user works with [LBU+05], or even users themselves [PPC01]). It also provides a *controlled* setup: first, as we can film all objects in front of the same backgrounds, we can generate a scenario in which clutter is widely uncorrelated from the object class. Second, if a static camera is used, we can obtain a close-to-perfect segmentation using background subtraction techniques (which is not possible for general scenes). This allows us to study how accurate object recognition could be if a perfect segmentation was available, and what influence the quality of segmentation has (by comparing a near-perfect segmentation with an error-prone one).

[10] http://delicious-monster.com/

Dataset and Setup The dataset for this experiment was manually generated: 12 books were chosen and presented to a static camera[11] at 12 indoor locations, obtaining 144 video clips of about 3 seconds length each. Sample frames from the dataset are illustrated in Figure 6.10. Strong changes of illumination and pose can be observed as well as significant clutter. Yet, despite these difficulties, this data represents a test case for which a state-of-the-art object recognition system is expected to work well. Frames were scaled to a resolution of 320×240 pixels and sampled at a framerate of 4 fps, obtaining $3-4$ views per object and location (frames not showing the object were omitted). SURF features [BTvG06] were extracted, obtaining about 600 patches on average per frame.

The system was tested using one-shot learning: one location is picked, and the videos taken at this location are used as training samples. Then, objects are recognized in the 132 clips taken at the other locations. For each video clip and object, a score is computed (see Equations (6.14) and (6.15)), and the object with the highest score is decided to appear in the clip. This experiment was repeated with all 12 locations as training data, obtaining an overall of $1,584$ object decisions. The error rate over these decisions is used as a performance measure.

Three setups of motion segmentation were tested: one not using segmentation at all, one using a near-perfect background subtraction, and one using the dynamic scene segmentation from Section 6.5 combining motion information with color models. Its parameters ($\alpha = 0.05, \beta = 6, \gamma = 2.5$, and $\eta = 0.5$) were estimated in a grid search minimizing segmentation error on a part of the dataset, and an affine motion model was used. Similar to Bouthemy [BF93], the segmentation result from previous frames (if there was a foreground region) was used as a starting point for segmentation. Images were scaled to a resolution of 160×120 pixels for segmentation. Sample segmentations are given in Figure 6.11. It can be seen that background subtraction gives close-to-perfect segmentations, while dynamic scene segmentation — which does not impose the strong assumption of a static background — tends to be inaccurate.

The recognition settings outlined in Section 6.6.1 were tested. First, a full patch search was tested, using a kd-tree for approximate nearest neighbor [PPC01]. The nearest neighbor ratio λ was varied such that $\lambda \in \{0.6, 0.7, 0.8, 0.9, 1.0\}$. Further, the approach was tested with and without a spatial refinement. Here, a tolerance of $\epsilon = 5$ pixels for deviations between model features and test features was validated to work best. Finally, full patch search was also compared with a faster approach discretizing patches to visual words.

[11] Canon Powershot S80

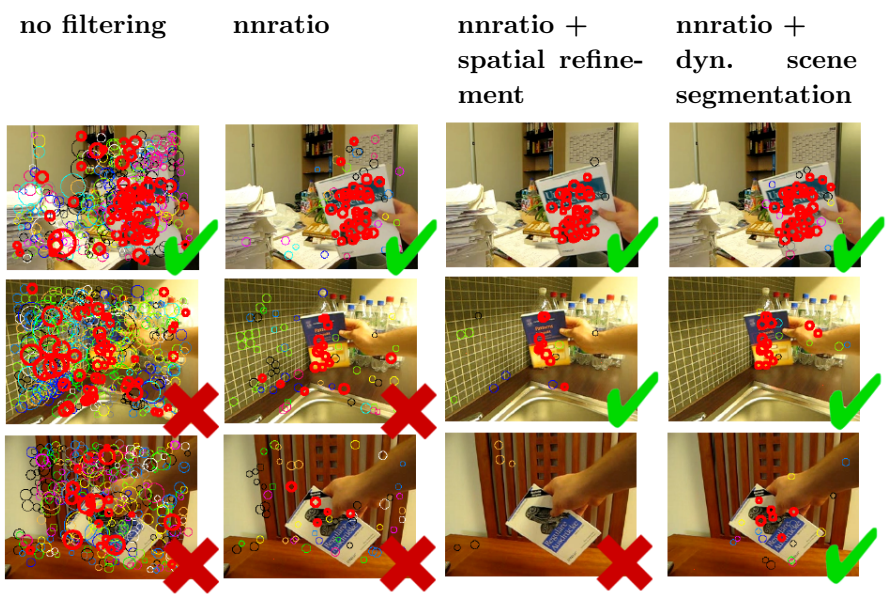
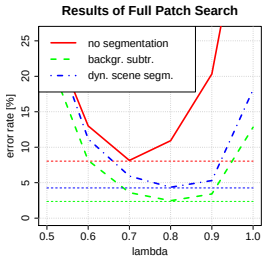


Figure 6.12: Sample recognition results using full patch search and different filtering techniques (correct votes are highlighted in red). Filtering by the nearest neighbor ratio ($\lambda = 0.9$, second column) reduces the influence of clutter to some extent. An additional spatial refinement (third column) can improve the accuracy of the match set, but sometimes discards patch groups entirely. A filtering by motion segmentation (rightmost column) gives the correct result in all cases.

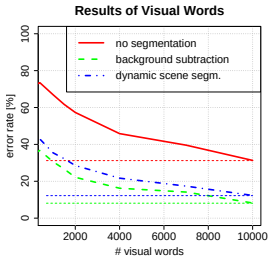
Results We start with an illustration of sample results in Figure 6.12. Three examples are shown, with patches colored according to the object they vote for. Correct votes are highlighted in red. The top row shows a simple case, where enough patches in the object region can be matched reliably, and an unfiltered voting using full patch search (first column) already gives the correct object decision. A further filtering by the nearest neighbor ratio ($\lambda = 0.9$, second column), by a refinement according to spatial constellation (third column), or by a motion-based segmentation (fourth column) increases the confidence of this object decision further. More difficult cases are illustrated in the second and third row. For both, a plain matching without filtering gives an incorrect result, and even the nearest neighbor ratio refinement does not lead to the correct object decision. A spatial refinement helps in one case but discards all positive object votes in the other. Only motion segmentation leads to the correct object decision in all cases.



(a)

error (full patch search)		
	spatial refinement	
segmentation	no	yes
no	8.14	8.21
backgr. subtr.	**2.46**	2.84
dyn. scene segm.	4.36	5.24

(b)



(c)

Figure 6.13: Quantitative Results of Experiment 1 ($n = 1,584$). (a) Results of full patch search. (b) A spatial refinement does not give any improvements over a filtering by the nearest neighbor ratio and motion segmentation. (c) For visual words voting, motion segmentation again provides strong performance improvements.

Quantitative results are given in Figure 6.13. First, in Figure 6.13(a), results for full patch search without spatial refinement are given. The error rate is plotted against the nearest neighbor ratio λ. It can be seen that recognition can be improved significantly by filtering according to the nearest neighbor ratio — for example, classification error is improved from 51.0% ($\lambda = 1$) to 8.1% ($\lambda = 0.7$) in the unsegmented case, which roughly corresponds to recognition rates reported for a commercial patch-based system [MPDB$^+$06].

However, when comparing results without segmentation (red) with a near-perfect segmentation (green) and dynamic scene segmentation (blue), we see that both segmentations help to reduce error further. For a slightly higher $\lambda = 0.8$ (since segmentation pre-filters patches, a less greedy rejection by nearest neighbor ratio turns out to be advantageous), the error rate for background subtraction is 2.5%. Dynamic scene segmentation does not reach this result, but still gives an improvement (4.4% error). Both improvements are significant according to a paired t-test over different training locations (level 98%).

We compare these results with a refinement by spatial constellation, for which parameters $\lambda = 0.9$ and $\epsilon = 5$ were obtained by a grid search optimization of recognition error. In Figure 6.13(b), it can be seen that this spatial refinement does not lead to any further improvements of recognition rate. A reason for this is indicated in Figure 6.12: while in some cases it helps to filter outliers, spatial refinement also tends to discard correct matches in some frames.

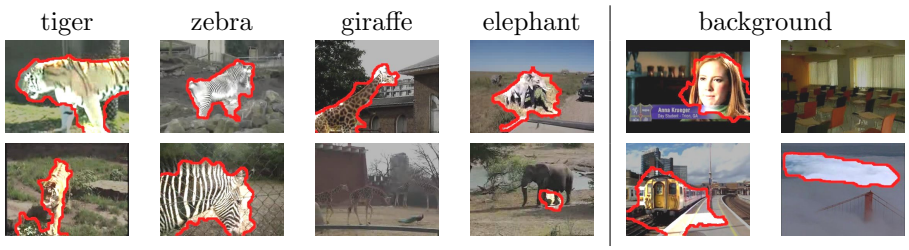


Figure 6.14: Sample segmentation results in Experiment 2. Foreground objects are segmented in many cases, though results are far from accurate. Typical sources of failure are slow motion (bottom giraffe) or a segmentation of body parts only (bottom elephant). Pictures from YouTube.

Finally, the visual words approach was evaluated, which achieves a more efficient matching by discretizing patches. In Figure 6.13(c), recognition error is plotted against the number of visual words. It can again be observed that motion segmentation leads to significant improvements: an error rate of 8.21% is reached by background subtraction, 12.31% for dynamic scene segmentation, and 31.31% without segmentation. Again, these improvements are significant (paired t-test, level 99%). Also, it can be seen that system accuracy increases strongly when using more clusters. The best result is achieved for a codebook size of $10,000$ (this cannot be increased much further, as the overall number of training features is about $13,000$).

6.6.3 Experiment 2: Concept Detection

Experiment 1 was conducted in a *controlled* setup in the sense that a static camera was used and that scene background was not correlated with the object class. In the following experiment, we will drop these assumptions and study the proposed framework in a real-world concept detection scenario, i.e. the system is applied to dynamic scenes, and background may be correlated with the object. As a test domain, the detection of animals in video databases was chosen. Similar experiments have previously been conducted by Ramanan et al. [RFB06], who presented a part-based model learned from video and validated improvements over baselines operating on plain images. Similar results will be described in the following for the proposed framework with motion segmentation.

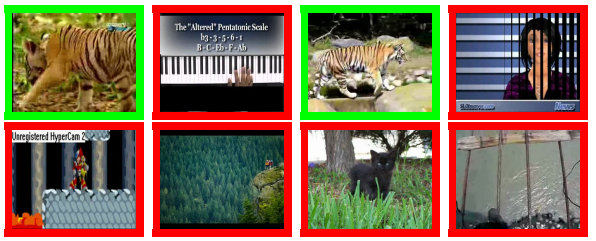
Dataset Experiments were run for four animal categories ("tiger", "zebra", "giraffe", "elephant"). Content downloaded from YouTube was used as test material for each category (and also for the background class). Videos were downloaded using a few manually defined queries: for example, for the concept "giraffe", queries like "giraffe zoo", "giraffe running", and "giraffe serengeti" were used. From the resulting video clips, small shots of $1-2$ seconds length were sampled. As our focus is on the influence of motion segmentation, this dataset was manually filtered with respect to a number of criteria:

- Only shots were accepted that show animals of sufficient size and visible to a sufficient extent.
- The animal is moving (for example, shots showing a tiger lying in the sun were discarded).
- Shots should not show multiple animals moving in different directions. This restriction is imposed by the current segmentation (which assumes a single foreground region to be given) and may be overcome in the future using an extension of motion segmentation to multiple regions (e.g., [KTZ05a]).
- Also, shots were discarded if they showed overlayed text or black bars at the top and bottom, which causes difficulties for motion segmentation.
- Finally, to avoid overfitting, only a single shot per YouTube clip was used.

This way, $29-50$ shots were sampled per animal category (160 shots total). This dataset was combined with a background class represented by $1,000$ shots sampled from the YouTube-22Concepts Dataset (see Section 3.6.1).

Setup The framework outlined in Section 6.6.1 was applied, with dynamic scene segmentation from Section 6.5. Parameters were optimized on a different dataset to $\alpha = 0.05, \beta = 6, \gamma = 1.5$, and $\eta = 0.5$. A constant motion model was used, and frames were sampled at a stepsize of 3 and smoothed with a Gaussian previous to motion segmentation.

While in Experiment 1 the accuracy of an object decision was measured, the object score is now used for a *ranking* of test content. For each animal category, a separate run was conducted in which animals from the category were classified versus the YouTube background class. Due to the limited number of positive examples, this evaluation was done in a leave-one-out fashion, i.e. each shot was scored by using the rest of the dataset as a training set. For efficiency reasons, only



(a)

avg. prec. (%)		
	segmented	
category	no	yes
tiger	15.6	**34.5**
zebra	27.2	**34.1**
giraffe	27.2	**42.5**
elephant	53.7	**55.3**
MAP	30.9	**41.6**

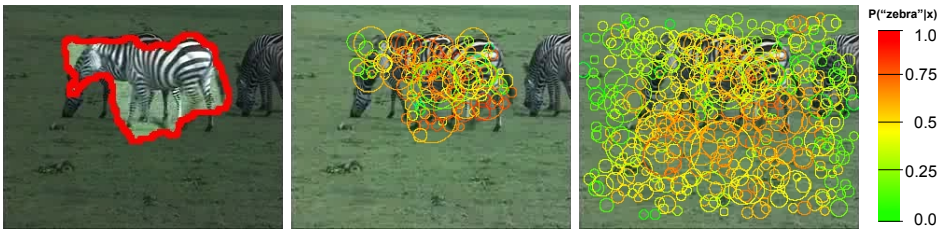
(b)

Figure 6.15: (a) The top 4 detection results for "tiger" with motion segmentation (top) and without (bottom). . (b) Quantitative Results on the YouTube-Animals Dataset ($n = 4,160$). Pictures from YouTube.

the *visual words* variant of the proposed framework was tested. Keyframes were sampled at regular steps of 6 frames, and SURF features [BTvG06] were extracted and mapped to a 5,000-dimensional visual codebook trained on the whole dataset. One global histogram of visual words was stored for the "animal" class and the "background" class, and matching was performed by computing the inner product of visual word histograms (see Equation (6.16)). A fusion over keyframes was done by accumulating votes from all frames of a shot.

Results Some sample results of motion segmentation are illustrated in Figure 6.14. It can be seen that in the majority of cases the animal in the foreground is detected correctly, but also that the segmentation is far from accurate. Typical sources of error are illustrated as well: the animal may be missed as a whole, which may happen if motion is too slow as in the bottom "giraffe" frame (in this case, the object score is set to $P(\text{"animal"}|V) = \frac{1}{2}$). Also, sometimes only body parts in strongest motion are segmented (like the legs of the bottom "elephant"). For the background class, other kinds of moving objects are segmented (like the face or the train). If no foreground motion takes place, a foreground region is sometimes hallucinated by the system.

Let us now study the combination of segmentation with patch-based recognition. Figure 6.15 illustrates the top 4 detection results for the concept "tiger" with segmentation and without. While in the unsegmented case no hits are found, the system with segmentation detects two tigers correctly. Note also that the other two detection results show dark vertical bars similar to tiger fur. Figure 6.15



(a) (b) (c)

Figure 6.16: The "zebra" recognizer on a YouTube sample shot. (a) Result of motion segmentation. (b,c) Result of recognition. Red patches indicate high scores, green patches low ones. Focusing on the object region (b) provides higher scores and thus gives a better result than recognition on the unsegmented frame (c). Picture from YouTube.

also provides quantitative results, where the average precision of object retrieval is compared with and without segmentation. It can be seen that for all animals, improvements are achieved by segmenting foreground objects. These are high for three of four animals, with "elephant" being the only exception. An in-depth inspection revealed that the segmentation of elephants does in fact fail more often than for the other animals — typically only certain body parts like legs or trunks are segmented. Overall, if combined with segmentation, a mean average precision of 41.6% is achieved, which is distinctly better than the system without segmentation (30.9%). This improvement is significant (sign test over rank improvement of positive samples, level 99%). A baseline using random guessing achieves 3.8%.

A sample recognition result is illustrated in Figure 6.16 for a "zebra" frame. Votes are visualized when using segmentation and when not, whereas red indicates a strong vote for "zebra" and green a strong vote for the background class. Two effects can be observed. First, votes from the background region tend to be noisy, also containing evidence for the background class. These votes are avoided by a segmentation of the test frame. Second, when comparing the votes within the object region, it can be seen that in the segmented case patches give a stronger vote for the "zebra" class. This is because a segmentation in training leads to a less noisy "zebra" model with stronger focus on the object itself.

6.7 Discussion

In this chapter, an approach for the automatic recognition of objects in video was presented. Its key contribution lies in combining patch-based recognition with motion segmentation. In this setup, frames are segmented into regions of coherent motion, and recognition is applied to these regions instead of on frame level. By applying motion segmentation this way, incorrect correspondences due to clutter can be reliably discarded. In two experiments — one related to the recognition of specific objects, one to concept detection — it has been demonstrated that adding segmentation information leads to significant improvements.

We have discussed previously that these results were not necessarily to be expected. Three potential arguments against an improvement by motion segmentation were given, namely an inherent robustness of patch-based recognition, the benefits of a correlated scene context, and inaccuracy of motion segmentation. Let us briefly discuss these issues with respect to the experimental results:

- First, patch-based methods have previously been shown to detect objects even in cluttered scenes [PHSZ07], as incorrect correspondences can be discarded based on feature appearance or position. Experimental results in this chapter confirm this robustness to some extent, but also show that some error remains, and that motion-based segmentation offers a more reliable way of filtering incorrect correspondences compared to patch appearance or spatial constellation. In a recognition experiment, error was reduced by ca. 50% using segmentation ($n = 1,584$).

- Second, one might expect that background — if correlated with the object to be recognized — can provide valuable context information [Tor03] and thus support recognition. In the two experiments presented, scene background played different roles: in Experiment 1, clutter was not correlated with the object class, as all objects were trained and recognized in the same scenes. As was to be expected, background did not give any helpful clues for recognition. In Experiment 2 — where the detection of animals in video databases was addressed — background was inherently correlated with the object class, as animals usually appear in zoos or in the wild. Again — despite this correlation — recognition was improved significantly, which indicates that segmentation serves as a stronger clue for recognition than context. This confirms results by Zhang et al. [ZMLS07], who studied the influence of clutter on object category recognition using image datasets from the PASCAL Challenge. Similar to the experiments conducted here, it was found that by

using only features from the manually segmented object region, recognition performance improved. Beyond this, our results also show that the same can be achieved by an *automatic* segmentation for video data.

- Third, motion segmentation is error-prone and limited. It should be kept in mind that the focus of this chapter was on situations where motion segmentations *can* be expected to work (i.e., showing a single object in motion). Here, motion segmentation, even if inaccurate, gave significant improvements. However, to employ segmentation in a truly unsupervised fashion for concept detection, several problems remain to be solved. First, situations with multiple moving objects may be addressed using appropriate motion segmentation techniques (e.g., [KTZ04]). This, however, leads to a *correspondence* problem: as we may face multiple regions per image, we do not know which ones are associated with the object of interest. Approaches to infer such correspondences have been developed in the image annotation literature [DBdFF02, KCdF04, YLP00] and are an option to overcome this problem.

  Also, motion segmentation will not help in cases where the object remains static, and all results so far have been based on the precondition of distinctive object motion. To overcome this problem, an iterative fitting procedure can be envisioned, in which alternately segmentations and object models are inferred. In this framework, motion-based segmentation could serve as an initial trigger for constructing a first object model only from scenes with moving objects. In later stages of learning, this object model can be used as an additional clue for top-down segmentation, such that objects can also be segmented and learned in other scenes where the object is static. One could also envision a semi-automatic learning framework: instead of annotating keyframes, users are presented frames together with segmentation results, and during annotation they can switch off segmentation or preserve it if appropriate.

Alternative directions of future work might include comparisons with other motion segmentation frameworks. So far, a *direct* approach has been followed, i.e. a dense segmentation of the scene has been inferred from the spatio-temporal derivatives and color statistics (see Section 6.5). Possible future directions of research might be to investigate sparse, feature-based approaches [TZ00]: as recognition in the proposed framework is entirely based on patches, it might be sufficient to perform a segmentation on basis of these sparse features. As strong arguments may be found in favor of both direct and feature-based methods [IA00, TZ00], it might

be interesting to compare both in a recognition framework. Similarly, another comparison could be done with the framework by Rothganger et al. [RLSP07], who cluster scenes into objects of coherent motion and infer an intermediate 3D representation of each object. Generally, our results demonstrate that such a 3D representation — while interesting for other applications — is not mandatory in a recognition framework, and that improvements of recognition can be achieved using a simpler (and computationally more efficient) 2D approach. Yet, it might be interesting to investigate how the proposed framework compares to a matching based on 3D scene representations.

Overall, the key results of this chapter are that despite progress in patch-based recognition, robustness with respect to clutter remains a key challenge, that, correspondingly, solving the segmentation problem can still be considered an important step towards a robust recognition, and that motion segmentation provides an appropriate way to do this.

Chapter 7

Discussion

Concept detection is an exciting field — as digital video databases grow at enormous rates [Jun09] and a precise manual indexing becomes infeasible, automatic video tagging can be attributed a good chance of becoming an integral part of future content-based video search technology. However, for it to gain traction, a fundamental scalability problem remains to be solved: thousands of concepts need to be learned, overfitting to training domains must be overcome, and systems must become flexible enough to catch up with users' information needs. This requires new strategies of bootstrapping visual learning beyond the manual annotation of limited datasets that constitutes the state of the art.

Therefore, the core concern of this thesis has been the question: can we learn to detect visual concepts in video with less supervision? To achieve this goal, we have turned towards web-based video sharing portals like YouTube. These services offer new sources of information that open great chances for concept learning.

First (and most obviously), vast amounts of visual content are available. We have seen in Chapter 3 that this content can be employed by concept detection systems, with experimental results indicating that YouTube-based detectors generalize comparably well to new domains as the ones trained on manually acquired annotations. Also, by adding YouTube content to other training sets, generalization capabilities can be improved.

Besides the scale of content, another appealing characteristic of portals like YouTube lies in the active participation of their users. Content is not only uploaded, but is also annotated, categorized, and debated about. This has allowed us to derive label information for concept learning — however, we have also seen that this information is of a different kind than the accurate annotations used

in concept detection before, as YouTube users give spontaneous, subjective, and coarse descriptions of their content. This has been shown to be a key problem with web-based training sets (Chapter 4), as significant amounts of material do not show the target concept. To overcome this problem, *relevance filtering* was proposed, which adapts the statistical models underlying concept detection such that unrelated content is filtered during system training. It could be seen that — by using this approach — the robustness of concept learning from web video could be improved.

Another valuable feature beyond content, tags, and descriptions is that users of portals like Flickr or YouTube sort their pictures into semantic categories. A novel way of exploiting this information has been suggested in Chapter 5, which is based on a combination of concept detection with style modeling from the domain of optical character recognition (OCR) [MB02, SN05]. The fact is used that the content to be annotated comes in groups sharing a coherent *style* (like TV shows, or snapshots taken over the same holiday trip). Based on evidence from the whole group, this style is matched with a previously learned web category, and a category-specific annotation model leads to an accurate concept detection. With this approach, significant performance improvements have been validated over a separate tagging of individual images.

Finally, we have addressed motion as an additional information source that is specific to video. A novel concept detection approach was presented that employs motion information for a segmentation of objects from the background. Recognition is then carried out on the level of the resulting object regions, which was shown to achieve an improved detection of concepts related to physical objects.

The experimental results presented in this thesis demonstrate that with the proposed framework, significant improvements can be achieved in terms of recognition performance and required supervision (and with it: scale and flexibility). It should also be noted that — besides the use of web-based information — another key aspect has been the design of proper statistical models. While previous approaches have usually cast concept detection as a standard supervised learning problem on labeled sample frames, the work in this thesis has been targeted at breaking out of this paradigm. Different information sources have been investigated, including information on context (Chapter 5), information on the reliability of labels (Chapter 4), and intra-frame segmentation information (Chapter 6). To make use of such information, a view of concept detection beyond standard supervised learning must be taken, and statistical modeling must be adapted.

Altogether, the contributions of this thesis can be combined in a novel approach for an efficient and widely unsupervised visual learning. An outline of this framework has already been given in the introduction (see Figure 1.2): the basis is an acquisition of training content from the web, which can be refined using motion segmentation or a filtering of non-relevant material. In parallel, category information can be used by learning several style-specific concept detectors instead of a global one. Though these processing steps can be applied independently (which has been the case in most experiments throughout this thesis), it should be noted that — if applied together — the different improvements can be expected to boost each other: for example, motion segmentation leads to better features by discarding the background, which can again simplify the identification of non-relevant material in relevance filtering. Correspondingly, one promising future direction along the proposed line of research is an integration of the different strategies.

Despite these achievements, this work can only be seen as a first step towards web-based concept learning. We have addressed some of the chances and challenges offered by user-tagged content, but have only started to understand what exactly we can learn from portals like Flickr and YouTube, how to cope with the enormous diversity of content, and how to automatically interpret its sparse and noisy tags and descriptions.

One remaining challenge is the "domain change problem", which refers to the fact that concept detection is often applied on different domains of video data (like TV) than it was trained on (here, web video). We have already seen in Chapter 3 – where YouTube-based detectors were evaluated on datasets of news and documentary TV from the TRECVID benchmark – that this domain switch causes severe problems, mainly because the appearance of target concepts can change significantly. Without doubt, when targeted at learning from web video we need to face this problem, and cross-domain analysis is one important direction of future work on web-based concept learning.

Another aspect that deserves more investigation is the enormous scale of web content, which might help to improve concept learning further by using much larger quantities of data. While the experiments in this thesis have in many cases already employed more training samples compared to manually acquired concept detection standard data, the full quantity of available material has not been exploited yet, and it is to be expected that — as sample size increases — tasks like relevance filtering and style selection will become easier. In this context, it might also be useful to learn from a variety of web-based information sources simultaneously, including video from YouTube as well as images from a web search or Flickr.

Finally, another chance to improve concept learning from the web is a limited amount of manual supervision. While this has been avoided deliberately in this thesis, at some point the capacity of a fully automatic system may be restricted by the diversity of content, even with more appropriate models and more data. Here, the key question is whether limited manual supervision can be integrated without sacrificing the scalability advantages of web content. The proposed framework offers several alternatives to include such extra information: users might be asked to refine training data by specifying more descriptive queries for downloading "good" YouTube content, or might provide a few thoroughly selected annotations to filter out non-relevant material. Other alternatives could be a manual grouping of content such that style information can be exploited, or a semi-automatic motion segmentation of objects from the background. Some recent work by the author of this thesis has already addressed this problem, and initial results indicate that the robustness with respect to label noise can in fact be improved by a low-cost semi-automatic refinement. Yet, important issues are far from solved, namely *how much* manual effort pays off best and *where* to best invest it — ultimately, to achieve concept detection that is both highly accurate *and* scalable, we will need to find the right level of manual supervision.

Appendix A

Test Concept Information

This section gives a description of test concepts used in Chapter 4. These definitions are related to the *visual* presence of a concept (for example, "desert" is associated with "scenes showing desert landscape"). Table A.1 provides definitions as well as information on how video data was downloaded from YouTube (i.e. what queries were made to the YouTube API).

Table A.1: Definitions and download information regarding the 10 test concepts used in Chapter 4

concept	description	YouTube Query	YouTube Category
basketball	Scenes showing people playing basketball. Includes streetball if recognizable as such.	basketball, basketball nba, basketball dunking, basketball best moves, basketball dunks	Sports
beach	Scenes showing a beach. Water does not have to be visible (if anything else qualifies the scene as showing a beach). Shots from a distance qualify as well, but only if the coastline is clearly a beach.	walk on the beach, beach sunbath, beach hawaii, beach panorama, beach malibu day	Travel&Places
cats	Scenes showing one or multiple cats. Closeups qualify as well as full body shots.	cats, cats funny, cats pets animals, cats playing, cats eating	Pets&Animals

desert	Scenes showing desert landscape. Panoramic shots involving significant amounts of sky are allowed (as long as some desert landscape is visible at the bottom). Things like plants, rocks, canyons, cars, etc. are allowed, but the landscape should show desert.	desert egypt, driving through desert, desert panorama, desert sahara, desert trip	Travel&Places
eiffeltower	Scenes showing the Eiffel Tower. Views from top of the tower qualify if you see a part of it, like parts of the steel construction. Night shots qualify. Closeups showing only parts of the steel construction qualify (if the tower can be identified) as well as panoramic shots from a distance. Shots with people in the foreground and the tower in the background count as well.	tour eiffel, eiffel tower, eiff.t. paris france, eiffelturm paris	Travel&Places
helicopter	Scenes showing a helicopter (airborne or on the ground). Views from inside the helicopter are allowed if they can be identified as such. Only instruments or the pilot are not sufficient. Shots of toy helicopters qualify as well.	helicopter, helicoptero, helicopter flying, helicopter landing	Autos&Vehicles
sailing	Scenes showing sailing ships/boats on the water/in the harbor. Panoramic views from onside a boat qualify if you see a part of the boat (like ropes or sails). Catamarans qualify as sailing ships, but surf boards or tankers do not (generally, everything with a sail qualifies).	sailing, sailing trip, sailing boat, sailing holiday, sailing mediterranean	Travel&Places,Sports

soccer	Shots showing a soccer match. Actions only weakly related to soccer do not qualify (like people doing soccer tricks in the street). Close-ups of players are allowed as well as global shots (if clearly identifiable as soccer). Soccer fields without action qualify as well. Shots of a cheering crowd do not qualify.	soccer bundesliga, soccer goals, soccer match, soccer game outdoor, fussball spiel	Sports
swimming	Scenes showing somebody swimming or a swimming pool (even if nobody is swimming inside it). Also includes swimming objects (fish, bottles).	swimming, swimming pool -clean, swimming technique, sw. competition, sw. olympics, sw. championship	Sports
tank	Scenes showing a tank, i.e. a heavily armored vehicle. Any scene qualifies if a part of the tank is visible such that the tank is identifiable. Other sorts of military ground vehicles qualify as well.	tanques, tank, tank battle, panzer, tank fire -flashpoint	Autos&Vehicles

Bibliography

[ABC+03] A. Amir, M. Berg, S. Chang, W. Hsu, G. Iyengar, C.-Y. Lin, M. Naphade, A. Natsev, C. Neti, H. Nock, J. Smith, B. Tseng, Y. Wu, and D. Zhang. IBM Research TRECVID-2003 Video Retrieval System. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2003.

[ACAB99] E. Ardizzone, M. La Cascia, A. Avanzato, and A. Bruna. Video Indexing using MPEG Motion Compensation Vectors. In *Proc. Int. Conf. on Multimedia Computing and Systems*, volume 2, pages 725–729, June 1999.

[ADDK99] Y. Avrithis, A. Doulamis, N. Doulamis, and S. Kollias. A Stochastic Framework for Optimal Key Frame Extraction from MPEG Video Databases. *Comput. Vis. Image Underst.*, 75(1-2):3–24, 1999.

[AdFDJ03] C. Andrieu, D. de Freitas, A. Doucet, and M. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1):5–43, 2003.

[Adi85] G. Adiv. Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(4):384–401, 1985.

[AET98] Y. Altunbasak, P. Eren, and A. Tekalp. Region-based Parametric Motion Segmentation using Color Information. *Graph. Models Image Process.*, 60(1):13–23, 1998.

[AKJ04] R. Akbani, S. Kwek, and N. Japkowicz. Applying Support Vector Machines to Imbalanced Datasets. In *Proc. Europ. Conf. Machine Learning*, pages 39–50, September 2004.

[ANV09] Anvato - ContentID Technology. available from `http://www.anvato.com/contentid-technology.php` (retrieved: Feb'09), February 2009.

[AQ07] S. Ayache and G. Quenot. TRECVID 2007 Collaborative Annotation using Active Learning. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2007.

[AQ08] S. Ayache and G. Quenot. Video Corpus Annotation using Active Learning. In *Proc. Europ. Conf. on Information Retrieval*, pages 187–198, March 2008.

[Auc07] E. Auchard. Flickr to Map the World's Latest Photo Hotspots. Reuters Group Limited; available from `http://www.reuters.com/article/technologyNews/idUSHO94233920071119` (retrieved: Feb'09), November 2007.

[BA96] M. Black and P. Anandan. The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields. *Comp. Vis. Img. Underst.*, 63(1):75–104, 1996.

[BAHH92] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical Model-Based Motion Estimation. In *Proc. Europ. Conf. Computer Vision*, pages 237–252, May 1992.

[Bal81] D. Ballard. Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[BB96] S. Beauchemin and J. Barron. The Computation of Optical Flow. *ACM Computing Surveys*, 27(3):433–467, 1996.

[BBC06] Britain is 'Surveillance Society'. in BBC News; available from `http://news.bbc.co.uk/1/hi/uk/6108496.stm` (retrieved: Feb'09), November 2006.

[BBC08] Picasa Refresh Brings Facial Recognition. in Techchrunch; available from `http://www.techcrunch.com/2008/09/02/picasa-refresh-brings-facial-recognition/` (retrieved: Feb'09), February 2008.

[BDF+03] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching Words and Pictures. *J. Mach. Learn. Res.*, 3:1107–1135, 2003.

[BF93] P. Bouthemy and E. François. Motion Segmentation and Qualitative Dynamic Scene Analysis from an Image Sequence. *Int. J. Comput. Vision*, 10(2):157–182, 1993.

[BF06] T. Berg and D. Forsyth. Animals on the Web. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1463–1470, June 2006.

[BK04] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(9), 2004.

[Bla92] M. Black. Combining Intensity and Motion for Incremental Segmentation and Tracking Over Long Image Sequences. In *Proc. European Conference on Computer Vision*, pages 485–493, May 1992.

[BLI09] Blinkx Video Search Technology. available from `http://www.blinkx.com/video-technology` (retrieved: Feb'09), February 2009.

[BM98] A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co-training. In *Proc. Ann. Conf. on Computational Learning Theory*, pages 92–100, July 1998.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition using Shape Context. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[BN94] H. Baird and G. Nagy. A Self-Correcting 100-Font Classifier. *Proc. SPIE — The International Society for Optical Engineering*, 2181:106–115, 1994.

[Bre92] T. Breuel. Fast Recognition using Adaptive Subdivisions of Transformation Space. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 445–51, June 1992.

[Bre93] T. Breuel. Recognition by Adaptive Subdivision of Transformation Space: Practical Experiences and Comparison with the Hough Transform. In *IEE Colloquium on Hough Transforms*, pages 7/1–4, May 1993.

[Bre96] T. Breuel. Finding Lines under Bounded Error. *Pattern Recognition*, 29(1):167–178, 1996.

[Bre01a] T. Breuel. Classification by Probabilistic Clustering. In *Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 1333–1336, May 2001.

[Bre01b] T.M. Breuel. Modeling the Sample Distribution for Clustering OCR. *Proc. SPIE - The International Society for Optical Engineering*, 4307:193–200, 2001.

[Bre03a] T. Breuel. An Algorithm for Finding Maximal Whitespace Rectangles at Arbitrary Orientations for Document Layout Analysis. In *Int. Conf. Document Analysis and Recognition*, August 2003.

[Bre03b] T. Breuel. On the Use of Interval Arithmetics in Geometric Branch and Bound Algorithms. *Pattern Recogn. Lett.*, 24(9-10):1375–1384, 2003.

[BTvG06] H. Bay, T. Tuytelaars, and L. van Gool. SURF: Speeded Up Robust Features. In *Proc. Europ. Conf. Computer Vision*, pages 404–417, May 2006.

[BU02] E. Borenstein and S. Ullman. Class Specific, Top-Down Segmentation. In *Proc. Europ. Conf. Computer Vision*, pages 639–641, May 2002.

[BUB09] D. Borth, A. Ulges, and T. Breuel. Active Relevance Filtering for Weakly Labeled Web Video. In *Proc. Int. Conf. on Multimedia (submitted for publication)*, October 2009.

[Bur98] C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[BWP00] M. Burl, M. Weber, and P. Perona. Unsupervised Learning of Models for Recognition. In *Proc. Europ. Conf. Computer Vision*, pages 18–32, June 2000.

[Can86] J. Canny. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[CCMV07] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised Learning of Semantic Classes for Image Annotation and Retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3):394–410, 2007.

[CEJ+07] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. Loui, and J. Luo. Large-scale Multimodal Semantic Concept Detection for Consumer Video. In *Proc. Int. Workshop Multimedia Information Retrieval*, pages 255–264, September 2007.

[CH05] M. Christel and A. Hauptmann. The Use and Utility of High-Level Semantic Features in Video Retrieval. In *Proc. Int. Conf. Image and Video Retrieval*, pages 134–144, July 2005.

[CHJ+06] S.-F. Chang, W. Hsu, W. Jiang, L. Kennedy, D. Xu, A. Yanagawa, and E. Zavesky. Columbia University TRECVID-2006 Video Search and High-Level Feature Extraction. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2006.

[CHJ+08] S.-F. Chang, J. He, Y.-G. Jiang, E. El Khoury, C.-W. Ngo, A. Yanagawa, and E. Zavesky. Columbia University/Video-CityU/IRIT TRECVID2008 High-Level Feature Extraction and Interactive Video Search. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2008.

[CHL+07] M. Campbell, A. Haubold, M. Liu, A. Natsev, J. Smith, J. Tesic, L. Xie, R. Yan, and J. Yang. IBM Research TRECVID-2007 Video Retrieval System. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2007.

[CJYZ07] S.-F. Chang, W. Jiang, A. Yanagawa, and E. Zavesky. Columbia University TRECVID2007 High-Level Feature Extraction. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2007.

[CL01] 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`.

[CLK97] J. Choi, S.-W. Lee, and S.-D. Kim. Spatio-temporal Video Segmentation using a Joint Similarity Measure. *IEEE Trans. Circuits and Systems for Video Technology*, 7(2):279–286, 1997.

[CLKH08] L. Cao, J. Luo, H. Kautz, and T. Huang. Annotating Collections of Photos using Hierarchical Event and Scene Models. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[CPCM08] M. Cristani, A. Perina, U. Castellani, and V. Murino. Geo-Located Image Analysis using Latent Representations. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[CR08] A. Llorente Coto and S. Rüger. Can a Probabilistic Image Annotation System be Improved using a Co-occurrence Approach? In *Proc. SAMT Workshop on Cross-Media Information Analysis and Retrieval*, December 2008.

[CS05] D. Cremers and S. Soatto. Motion Competition: A Variational Approach to Piecewise Parametric Motion Segmentation and Shape Regularization. *Int. J. Comput. Vision*, 62(3):249–265, 2005.

[CSZ06] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-supervised Learning*. MIT Press, Cambridge, MA, 2006.

[DBdFF02] P. Duygulu, K. Barnard, J. de Freitas, and D. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In *Proc. Europ. Conf. Computer Vision*, pages 97–112, May 2002.

[Del02] F. Dellaert. The Expectation Maximization Algorithm. Technical Report GIT-GVU-02-20, Georgia Institute of Technology, 2002.

[DHS00] R. Duda, P. Hart, and D. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[DJLW07] R. Datta, D. Joshi, J. Li, and J. Wang. Tagging over Time: Real-world Image Annotation by Lightweight Meta-Learning. In *Proc. Int. Conf. on Multimedia*, pages 393–402, September 2007.

[DKN05] T. Deselaers, D. Keysers, and H. Ney. Discriminative Training for Object Recognition using Image Patches. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 157–162, June 2005.

[DKN08] T. Deselaers, D. Keysers, and H. Ney. Features for Image Retrieval: an Experimental Comparison. *Information Retrieval*, 11(2):77–107, 2008.

[DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[DPN08] T. Deselaers, L. Pimenidis, and H. Ney. Bag-of-Visual-Words Models for Adult Image Classification and Filtering. In *Proc. Int. Conf. Pattern Recognition*, pages 1–4, December 2008.

[DT05] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 886–893, June 2005.

[DUBW09] M. Duan, A. Ulges, T. Breuel, and X.-Q. Wu. Style Modeling for Tagging Personal Photo Collections. In *Proc. Int. Conf. Image and Video Retrieval*, July 2009.

[DZ07] D. Ding and D. Zhang. Probabilistic Model Supported Rank Aggregation for the Semantic Concept Detection in Video. In *Proc. Int. Conf. Image and Video Retrieval*, pages 587–594, July 2007.

[Elk03] C. Elkan. Using the Triangle Inequality to Accelerate KMeans. In *Proc. Int. Conf. Machine Learning*, pages 147–153, August 2003.

[Enk91] W. Enkelmann. Obstacle Detection by Evaluation of Optical Flow Fields from Image Sequence. *Image and Vision Computing*, 9:160–168, 1991.

[EVGW+07] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. Technical report, PASCAL Challenge Workshop. available from: `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/workshop.html`, October 2007.

[EVGW+08] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results, October 2008.

[EZWvG06] M. Everingham, A. Zisserman, C. Williams, and L. van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. Technical report, PASCAL Challenge Workshop. available from: `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2006/workshop.html`, May 2006.

[FB81] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FB04] F. Fraundorfer and H. Bischof. Evaluation of Local Detectors on Non-Planar Scenes. In *28th OAGM/AAPR Workshop*, pages 125–132, June 2004.

[FdW05] D. Farin and P. de With. Evaluation of a Feature-Based Global-Motion Estimation System. In *Conf. on Visual Communications and Image Processing*, pages 1331–1342, July 2005.

[FFFPZ05] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning Object Categories from Google's Image Search. *Computer Vision*, 2:1816–1823, 2005.

[FFP05] L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 524–531, June 2005.

[FM08] S. Feng and R. Manmatha. A Discrete Direct Retrieval Model for Image and Video Retrieval. In *Proc. Int. Conf. Image and Video Retrieval*, pages 427–436, July 2008.

[FML04] S. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli Relevance Models for Image and Video Annotation. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1002–1009, June 2004.

[FP02] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Prentice Hall, 2002.

[FPZ03] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 264–271, June 2003.

[FPZ05] R. Fergus, P. Perona, and A. Zisserman. A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 380–387, June 2005.

[FTG06] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous Object Recognition and Segmentation by Image Exploration. In *Toward Category-Level Object Recognition*, pages 145–169. Springer-Verlag New York, Inc., 2006.

[GAB05] M. Galun, A. Apartsin, and R. Basri. Multiscale Segmentation by Combining Motion and Intensity Cues. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 256–263, June 2005.

[GMH+07] Z. Gu, T. Mei, X.-S. Hua, J. Tang, and X. Wu. Multi-layer Multi-instance Kernel for Video Concept Detection. In *Proc. Int. Conf. on Multimedia*, pages 349–352, September 2007.

[GNC+08] A. Gallagher, C. Neustaedter, L. Cao, J. Luo, and T. Chen. Image Annotation using Personal Calendars as Context. In *Proc. Int. Conf. on Multimedia*, pages 681–684, October 2008.

[GOO09] Google Advanced Image Search. `http://images.google.com/advanced_image_search` (retrieved: Feb'09), Feburary 2009.

[GW02] R. Gonzalez and R. Woods. *Digital Image Processing (2nd Edition)*. Prentice Hall, January 2002.

[GY08] U. Gargi and J. Yagnik. Solving the Label Resolution Problem in Supervised Video Content Classification. In *Proc. Int. Conf. on Multimedia Retrieval*, pages 276–282, October 2008.

[Han02] A. Hanjalic. Shot-boundary Detection: Unraveled and Resolved? *IEEE Trans. Circuits and Systems for Video Technology*, 12(2):90–105, 2002.

[HCL03] C. Hsu, C. Chang, and C. Lin. A Practical Guide to Support Vector Classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, 2003.

[Hig06] A. Highfield. Digital Homes: On demand or over hyped? Session at the Media Guardian Edinburgh International Television Festival. in BBC Press; available from `http://www.bbc.co.uk/pressoffice/speeches/stories/highfield_edinburgh.shtml` (retrieved: Feb'09), August 2006.

[HL04] S. Helmer and D. Lowe. Object Class Recognition with Many Local Features. In *Proc. Int. Conf. Computer Vision and Pattern Recognition Workshop*, pages 187–195, June 2004.

[HM00] R. Hammoud and R. Mohr. A Probabilistic Framework of Selecting Effective Key Frames for Video Browsing and Indexing. In *Proc.*

Int. Workshop on Real-Time Image Sequence Analysis, pages 79–88, August 2000.

[HN07] A. Haubold and M. Naphade. Classification of Video Events using 4-dimensional Time-Compressed Motion features. In *Proc. Int. Conf. Image and Video Retrieval*, pages 178–185, July 2007.

[HN08] A. Haubold and A. Natsev. Web-based Information Content and its Application to Concept-based Video Retrieval. In *Proc. Int. Conf. Image and Video Retrieval*, pages 437–446, July 2008.

[HOdJ07] M. Huijbregts, R. Ordelman, and F. de Jong. Annotation of Heterogeneous Multimedia Content using Automatic Speech Recognition. In *Proc. Int. Conf. Semantics and Digital Media Technology*, pages 78–90, December 2007.

[Hof01] T. Hofmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196, 2001.

[HS80] B. Horn and B. Schunck. Determining Optical Flow. Technical report, Massachusetts Institute of Technology, 1980.

[HS88] C. Harris and M. Stevens. A Combined Corner and Edge Detector. In *Proc. 4th Alvey Vision Conference*, pages 147–151, May 1988.

[Hub74] P. Huber. *Robust Statistics*. Wiley-Interscience, 1974.

[HYL07] A. Hauptmann, R. Yan, and W. Lin. How many High-Level Concepts will Fill the Semantic Gap in News Video Retrieval? In *Proc. Int. Conf. Image and Video Retrieval*, pages 627–634, July 2007.

[HZ99] A. Hanjalic and H. Zhang. An Integrated Scheme for Automated Video Abstraction Based on Unsupervised Cluster-Validity Analysis. *IEEE Trans. Circuits and Systems for Video Technology*, 9(8):1280–1289, 1999.

[IA00] M. Irani and P. Anandan. About Direct Methods. In *Proc. Int. Workshop on Vision Algorithms*, pages 267–277, September 2000.

[Inc09] Cisco Systems Inc. Cisco Visual Networking Index: Forecast and Methodology, 2008-2013. available from `http://www.cisco.com` (retrieved: June'09), June 2009.

[IRP94] M. Irani, B. Rousso, and S. Peleg. Computing Occluding and Transparent Motions. *Int. J. Comput. Vision*, 12(1):5–16, 1994.

[JB93] A. Jepson and M. Black. Mixture Models for Optical Flow Computation. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 760–767, June 1993.

[JCL07] W. Jiang, S.-F. Chang, and A. Loui. Context-Based Concept Fusion with Boosted Conditional Random Fields. In *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, April 2007.

[JDS08a] H. Jegou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *Proc. Europ. Conf. Computer Vision*, volume I, pages 304–317, October 2008.

[JDS08b] H. Jegou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometry Consistency for Large Scale Image Search - Extended Version. Technical report, INRIA, RR 6709, 2008.

[JF01] N. Jojic and B. Frey. Learning Flexible Sprites in Video Layers. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 199–206, December 2001.

[JH99] A. Johnson and M. Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[JM04] J. Jeon and R. Manmatha. Using Maximum Entropy for Automatic Image Tagging. In *Proc. Int. Conf. Image and Video Retrieval*, pages 24–32, July 2004.

[JNY07] Y.-G. Jiang, C.-W. Ngo, and J. Yang. Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval. In *Proc. Int. Conf. Image and Video Retrieval*, pages 494–501, July 2007.

[Joa99] T. Joachims. Transductive Inference for Text Classification using Support Vector Machines. In *Int. Conf. Machine Learning*, pages 200–209, June 1999.

[JR02] M. Jones and J. Rehg. Statistical Color Models with Application to Skin Detection. *Int. J. Comput. Vision*, 46(1):81–96, January 2002.

[Jun09] R. Junee. Zoinks! 20 Hours of Video Uploaded Every Minute! The YouTube Blog; available from `http://www.youtube.com/blog?entry=on4EmafA5MA` (retrieved: May'09), May 2009.

[JWZ06] N. Jojic, J. Winn, and L. Zitnick. Escaping Local Minima Through Hierarchical Model Selection: Automatic Object Discovery, Segmentation, and Tracking of Objects. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 117–124, June 2006.

[KÖ3] T. Kölsch. Local Features for Image Classification. Technical report, Diploma Thesis, RWTH Aachen, 2003.

[KB01] T. Kadir and M. Brady. Saliency, Scale and Image Description. *Int. J. Comput. Vis.*, 45(2):83–105, 2001.

[KCdF04] H. Kück, P. Carbonetto, and N. de Freitas. A Constrained Semi-supervised Learning Approach to Data Association. In *Proc. Europ. Conf. Computer Vision*, pages 1–12, May 2004.

[KCK06] L. Kennedy, S.-F. Chang, and I. Kozintsev. To Search or to Label?: Predicting the Performance of Search-based Automatic Image Classifiers. In *Int. Workshop Multimedia Information Retrieval*, pages 249–258, October 2006.

[KCM04] E.-Y. Kang, I. Cohen, and G. Medioni. A Layer Extraction System based on Dominant Motion Estimation and Global Registration. In *Proc. Int. Conf. on Multimedia and Expo*, pages 551–554, June 2004.

[KDB07] D. Keysers, T. Deselaers, and T. Breuel. Optimal Geometric Matching for Patch-Based Object Detection. *Electr. Letters on Comp. Vis. Img. Anal.*, 6:44–54, 2007.

[KHDM98] J. Kittler, M. Hatef, R. Duin, and J. Matas. On Combining Classifiers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[KHN+06] L. Kennedy, A. Hauptmann, M. Naphade, J. Smith, and S.-F. Chang. LSCOM Lexicon Definitions and Annotations Version 1.0. Technical report, Columbia University, 2006.

[KO05] W. Kraaij and P. Over. TRECVID-2005 High-Level Feature Task: Overview. In *Proc. TRECVID Workshop (slides available from: http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html)*, November 2005.

[KO06] W. Kraaij and P. Over. TRECVID-2006 High-Level Feature Task: Overview. In *Proc. TRECVID Workshop (slides available from: http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html)*, November 2006.

[KO07] W. Kraaij and P. Over. TRECVID-2007 High-Level Feature Task: Overview. In *Proc. TRECVID Workshop (slides available from: http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html)*, November 2007.

[KO08] W. Kraaij and P. Over. TRECVID-2008 High-Level Feature Task: Overview. In *Proc. TRECVID Workshop (slides available from: http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html)*, November 2008.

[KOO09] kooaba Mobile Visual Search: point - snap - find. available from `www.kooaba.com/technology/` (retrieved: Feb'09), February 2009.

[KRB01] G. Kühne, S. Richter, and M. Beier. Motion-based Segmentation and Contour-based Classification of Video Objects. In *Proc. Int. Conf. on Multimedia*, pages 41–50, September 2001.

[KS01] S. Khan and M. Shah. Object Based Segmentation of Video using Color, Motion and Spatial Information. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 746–751, December 2001.

[KTZ04] M. Kumar, P. Torr, and A. Zisserman. Extending Pictorial Structures for Object Recognition. In *Proc. British Machine Vision Conf.*, pages 789–798, September 2004.

[KTZ05a] P. Kumar, P. Torr, and A. Zisserman. Learning Layered Motion Segmentations of Video. In *Proc. Int. Conf. Computer Vision*, pages 33–40, October 2005.

[KTZ05b] P. Kumar, P. Torr, and A. Zisserman. OBJ CUT. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 18–25, June 2005.

[KTZ08] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning Layered Motion Segmentations of Video. *Int. J. Comput. Vision*, 76(3):301–319, 2008.

[LBU+05] C. Lampert, T. Braun, A. Ulges, D. Keysers, and T. Breuel. Oblivious Document Capture and Real-Time Retrieval. In *Proc. Int. Workshop Camera Based Document Analysis and Recognition*, pages 79–86, August 2005.

[LCZ+06] X. Li, L. Chen, L. Zhang, F. Lin, and W.-Y. Ma. Image Annotation by Large-scale Content-based Image Retrieval. In *Proc. Int. Conf. on Multimedia*, pages 607–610, October 2006.

[Lew98] D. Lewis. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Proc. Europ. Conf. Machine Learning*, pages 4–15, April 1998.

[LG94] D. Lewis and W. Gale. A Sequential Algorithm for Training Text Classifiers. In *Proc. Int. Conf. Research and Development in Information Retrieval*, pages 3–12, July 1994.

[LH02] W.-H. Lin and A. Hauptmann. News Video Classification using SVM-based Multimodal Classifiers and Combination Strategies. In *Proc. Int. Conf. on Multimedia*, pages 323–326, December 2002.

[Lie01] R. Lienhart. Reliable Transition Detection in Videos: A Survey and Practitioner's Guide. *Int. J. of Img. and Graph.*, 1(3):469–286, 2001.

[Lin98] T. Lindeberg. Feature Detection with Automatic Scale Selection. *Int. J. Comput. Vis.*, 30(2):77–116, 1998.

[LK81] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Int. Joint Conf. Artificial Intelligence*, pages 674–679, August 1981.

[LLM03] J. Leon, V. Lavrenko, and R. Manmatha. Automatic Image Annotation and Retrieval using Cross-media Relevance Models. In *Proc. Int. SIGIR Conf. Research and Development in Information Retrieval*, pages 119–126, July 2003.

[LLS04] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *Proc.*

ECCV 04, Workshop Stat. Learn. Comp. Vis., pages 17–32, May 2004.

[LMJ04] V. Lavrenko, R. Manmatha, and J. Jeon. A Model for Learning the Semantics of Pictures. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[Low99] D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Int. Conf. Computer Vision*, pages 1150–1157, September 1999.

[Low04] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.

[LS07] B. Leibe and B. Schiele. Robust Object Detection by Interleaving Categorization and Segmentation. *Int. J. Comp. Vis.*, 77(1-3):259–289, 2007.

[LSC] LSCOM - Large-scale Concept Ontology for Multimedia. available from: `http://www.lscom.org` (retrieved: Aug'08).

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 2169–2178, June 2006.

[LSS05] Y. Li, J. Sun, and H.-Y. Shum. Video Object Cut and Paste. *ACM Trans. Graph.*, 24(3):595–600, 2005.

[LSW08] X. Li, C. Snoek, and M. Worring. Learning Tag Relevance by Neighbor Voting for Social Image Retrieval. In *Proc. Int. Conf. on Multimedia Information Retrieval*, pages 180–187, October 2008.

[Luc05] S. Lucas. Text Locating Competition Results. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 80–85, August 2005.

[LW08] J. Li and J. Wang. Real-time Computerized Annotation of Pictures. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

[LWFF07] L.-J. Li, G. Wang, and L. Fei-Fei. OPTIMOL: automatic Object Picture collecTion via Incremental MOdel Learning. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 57–64, June 2007.

[M. 95] M. Flickner et al. Query by Image and Video Content: The QBIC System. *Computer*, 28(9):23–32, 1995.

[MB02] C. Mathis and T. Breuel. Classification using a Hierarchical Bayesian Approach. In *Int. Conf. Pattern Recognition*, pages 103–106, August 2002.

[MCMP02] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *Proc. British Machine Vision Conf.*, pages 384–393, September 2002.

[MCSM07] H. Marin Castro, E. Sucar, and E. Morales. Automatic Image Annotation using a Semi-supervised Ensemble of Classifiers. In *Proc. Iberoamerican Congress on Pattern Recognition*, pages 487–495, November 2007.

[MFTM01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *Proc. Int. Conf. Computer Vision*, pages 416–423, July 2001.

[MGP04] F. Monay and D. Gatica-Perez. PLSA-based Image Annotation: Constraining the Latent Space. In *Proc. Int. Conf. on Multimedia*, pages 348–351, October 2004.

[MHYL07] T. Mei, X.-S. Hua, L. Yang, and S. Li. VideoSense: Towards Effective Online Video Advertising. In *Proc. Int. Conf. on Multimedia*, pages 1075–1084, July 2007.

[Mik03] K. Mikolajczyk. A Performance Evaluation of Local Descriptors. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 257–263, June 2003.

[MLS06] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple Object Class Detection with a Generative Model. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 26–36, June 2006.

[MMMP02] H. Müller, S. Marchand-Maillet, and T. Pun. The Truth about Corel - Evaluation in Image Retrieval. In *Proc. Int. Conf. on Image and Video Retrieval*, pages 38–49, July 2002.

[MOVY01] B. Manjunath, J.-R. Ohm, V. Vasuvedan, and A. Yamada. Color and Texture Descriptors. *IEEE Trans. Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

[MP05] P. Moreels and P. Perona. Evaluation of Features Detectors and Descriptors Based on 3D Objects. In *Proc. Int. Conf. Computer Vision*, volume 1, pages 800–807, July 2005.

[MPDB+06] M. Munich, P. Pirjanian, E. Di Bernardo, L. Goncalves, N. Karlsson, and D. Lowe. SIFT-ing through Features with ViPR. *IEEE Robotics & Automation Magazine*, 13(3):72–77, 2006.

[MPN08] N. Morsillo, C. Pal, and R. Nelson. Semi-supervised Visual Scene and Object Analysis from Web Images and Text. In *Scene Understanding Symposium*, February 2008.

[MR98] O. Maron and A. Rathan. Multiple-Instance Learning for Natural Scene Classification. In *Proc. Int. Conf. Machine Learning*, pages 341–349, July 1998.

[MRY06] P. Mundur, Y. Rao, and Y. Yesha. Keyframe-based Video Summarization using Delaunay Clustering. *Int. J. Digit. Libr.*, 6(2):219–232, 2006.

[MS04] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *Int. J. Comput. Vis.*, 60(1):63–86, 2004.

[MTO99] Y. Mori, T. Takahashi, and R. Oka. Image-to-Word Transformation based on Dividing and Vector Quantizing Images with Words. In *Proc. Int. Workshop Multimedia Intelligent Storage and Retrieval Management*, December 1999.

[Mun06] J. Mundy. Object Recognition in the Geometric Era: A Retrospective. In *Toward Category-Level Object Recognition*, pages 3–28. Springer-Verlag New York, Inc., 2006.

[MW98] R. Mech and M. Wollborn. A Noise Robust Method for 2D Shape Estimation of Moving Objects in Video Sequences Considering a Moving Camera. *Signal Process.*, 66(2):203–217, 1998.

[MWH+08] T. Mei, Y. Wang, X.-S. Hua, S. Gong, and S. Li. Coherent Image Annotation by Learning Semantic Distance. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, October 2008.

[MZ03] Y. Ma and H. Zhang. Motion Pattern-based Video Classification and Retrieval. *EURASIP J. Appl. Signal Process.*, 2003(1):199–208, 2003.

[NaSN96] S. Nayar and H. Murase ad S. Nene. *Parametric Appearance Representation*, pages 131–160. Oxford Univ. Press, 1996.

[NGP08] R. Negoescu and D. Gatica-Perez. Analyzing Flickr Groups. In *Proc. Int. Conf. Image and Video Retrieval*, pages 417–426, July 2008.

[NH01] M. Naphade and T. Huang. A Probabilistic Framework for Semantic Video Indexing, Filtering, and Retrieval. *IEEE Trans. Multimedia*, 3(1):141–151, 2001.

[NJT06] E. Nowak, F. Jurie, and B. Triggs. Sampling Strategies for Bag-of-Features Image Classification. In *Proc. Europ. Conf. Computer Vision*, pages 490–503, May 2006.

[NKK+05] M. Naphade, L. Kennedy, J. Kender, S.-F. Chang, J. Smith, P. Over, and A. Hauptmann. A Light Scale Concept Ontology for Multimedia Understanding for TRECVID 2005. Technical report, IBM Research Division, 2005.

[NLM99] K. Nigam, J. Lafferty, and A. Mccallum. Using Maximum Entropy for Text Classification. In *Proc. IJCAI Worksh. Mach. Learn. for Information Filtering*, pages 61–67, July 1999.

[NNT05] A. Natsev, M. Naphade, and J. Tesic. Learning the Semantics of Multimedia Queries and Concepts from a Small Number of Examples. In *Proc. Int. Conf. on Multimedia*, pages 598–607, November 2005.

[NS04] M. Naphade and J. Smith. On the Detection of Semantic Concepts at TRECVID. In *Proc. Int. Conf. on Multimedia*, pages 660–667, December 2004.

[NS06] D. Nister and H. Stewenius. Scalable Recognition with a Vocabulary Tree. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 2161–2168, June 2006.

[NST+06] M. Naphade, J. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-Scale Concept Ontology for Multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006.

[NWD00] H. Nguyen, M. Worring, and A. Dev. Detection of Moving Objects in Video using a Robust Motion Similarity Measure. *IEEE Trans. Image Processing*, 9(1):137–141, 2000.

[NYGMP05] M. Naaman, R. Yeh, H. Garcia-Molina, and A. Paepcke. Leveraging Context to Resolve Identity in Photo Albums. In *Proc. Joint Conf. Digital Libraries*, pages 178–187, June 2005.

[OAKS07] P. Over, G. Awad, W. Kraaij, and A. Smeaton. TRECVID 2007 - An Overview. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2007.

[OB07] B. Ommer and J. Buhmann. Learning the Compositional Nature of Visual Objects. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[OIKS05] P. Over, T. Ianeva, W. Kraaij, and A.F. Smeaton. TRECVID 2005 - An Overview. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2005.

[Pat08] F. Patalong. Nur falsch ist wirklich echt. Spiegel Online (German). available from `http://www.spiegel.de/netzwelt/web/0,1518,436070,00.html` (retrieved: Sep'08), September 2008.

[PBE+06] J. Ponce, T. Berg, M. Everingham, D. Forsyth, M. Hebert, S. Lazebnik, M. Marszałek, C. Schmid, C. Russell, A. Torralba, C. Williams, J. Zhang, and A. Zisserman. Dataset Issues in Object Recognition. In *Towards Category-Level Object Recognition*, pages 29–48. Springer-Verlag New York, Inc., 2006.

[PCD08] N. Pinto, D. Cox, and J. Dicarlo. Why is Real-World Visual Object Recognition Hard? *PLoS Computational Biology*, 4(1):e27+, 2008.

[PCI+07] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[PHSZ07] J. Ponce, M. Hebert, C. Schmid, and A. Zisserman. *Toward Category-Level Object Recognition*. Springer-Verlag New York, Inc., 2007.

[Pop94] A. Pope. Model-Based Object Recognition: A Survey of Recent Research. Technical Report TR-94-04, University of British Columbia, Computer Science Department, 1994.

[PPC01] R. Paredes and A. Perez-Cortes. Local Representations and a Direct Voting Scheme for Face Recognition. In *Proc. Workshop on Pattern Rec. and Inf. Systems*, pages 71–79, July 2001.

[PV02] R. Piroddi and T. Vlachos. Multiple-Feature Spatiotemporal Segmentation of Moving Sequences using a Rule-based Approach. In *Proc. British Machine Vision Conf.*, pages 353–362, September 2002.

[QMO+07] P. Quelhas, F. Monay, J. Odobez, D. Gatica-Perez, and T. Tuytelaars. A Thousand Words in a Scene. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(9):1575–1589, 2007.

[RAAKR05] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image Change Detection Algorithms: A Systematic Survey. *IEEE Trans. Image Processing*, 14(3):294–307, 2005.

[Ren08] X. Ren. Local Grouping for Optical Flow. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[RFB06] D. Ramanan, D. Forsyth, and K. Barnard. Building Models of Animals from Video. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(8):1319–1334, 2006.

[RJB06] H. Rowley, Y. Jing, and S. Baluja. Large Scale Image-Based Adult-Content Filtering. In *Int. Conf. Comp. Vis. Theory and Applications*, pages 290–296, February 2006.

[RL03] I. Ruthven and M. Lalmas. A Survey on the Use of Relevance Feedback for Information Access Systems. *Knowl. Eng. Rev.*, 18(2):95–145, 2003.

[RLSP06] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D Object Modeling and Recognition from Photographs and Image Sequences. In *Toward Category-Level Object Recognition*, pages 105–126. Springer-Verlag New York, Inc., 2006.

[RLSP07] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving

Objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3):477–491, 2007.

[Rot08] P. Roth. Survey of Appearance-based Methods for Object Recognition. Technical Report ICG-TR-01/08, Computer Graphics & Vision, TU Graz, 2008.

[RSSM01] H. Richter, A. Smolic, B. Stabernack, and E. Müller. Real Time Global Motion Estimation for an MPEG-4 Video Encoder. In *Proc. Picture Coding Symposium*, pages 401–404, April 2001.

[RvBKB08] M. Renn, J. van Beusekom, D. Keysers, and T. Breuel. Automatic Image Tagging using Community-Driven Online Image Databases. In *Proc. Int. Workshop Adaptive Multimedia Retrieval*, June 2008.

[SC06] T. Schoenemann and D. Cremers. Near Real-Time Motion Segmentation using Graph Cuts. In *Proc. DAGM-Symposium*, pages 455–464, September 2006.

[Sch78] G. Schwarz. Estimating the Dimension of a Model. *Ann. of Stat.*, 2(6):461–464, 1978.

[SCZ07] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting Image Databases from the Web. In *Proc. Int. Conf. Computer Vision*, pages 1–8, October 2007.

[Set09] B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[SHH+07] C. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring. Adding Semantics to Detectors for Video Retrieval. *IEEE Trans. Multimedia*, 9(5):975–986, 2007.

[Sib08] A. Sibiryakov. Estimating Inter-Frame Parametric Dominant Motion at 1000fps Rate. In *Proc. Int. Conf. on Consumer Electronics*, pages 1–2, January 2008.

[SJL+06] S. Sav, G. Jones, H. Lee, N. O'Connor, and A. Smeaton. Interactive Experiments in Object-based Retrieval. In *Proc. Int. Conf. Image and Video Retrieval*, pages 1–10, July 2006.

[SM00] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[SMB00] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of Interest Point Detectors. *Int. J. Comput. Vis.*, 37(2):151–172, 2000.

[Sme05] A. Smeaton. Large Scale Evaluations of Multimedia Information Retrieval: The TRECVid Experience. In *Proc. Int. Conf. Image and Video Retrieval*, pages 11–17, July 2005.

[Sme07] A. Smeaton. Techniques Used and Open Challenges to the Analysis, Indexing and Retrieval of Digital Video. *Inf. Syst.*, 32(4):545–559, 2007.

[SMH04] F. Souvannavong, B. Merialdo, and B. Huet. Improved Video Content Indexing by Multiple Latent Semantic Analysis. In *Proc. Int. Conf. Image and Video Retrieval*, pages 483–490, July 2004.

[Smo01] A. Smolic. *Globale Bewegungsbeschreibung und Video Mosaiking unter Verwendung parametrischer 2-D Modelle, Schaetzverfahren und Anwendungen* (German). PhD thesis, RWTH Aachen, Germany, 2001.

[SN05] P. Sarkar and G. Nagy. Style Consistent Classification of Isogenous Patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(1):88–98, 2005.

[Sno07] Snoek, C. et al. The MediaMill TRECVID 2007 Semantic Video Search Engine. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2007.

[SOK06] A. Smeaton, P. Over, and W. Kraaij. Evaluation Campaigns and TRECVid. In *Int. Workshop Multimedia Information Retrieval*, pages 321–330, October 2006.

[SOU09] Netherlands Institute for Sound and Vision. available from `http://instituut.beeldengeluid.nl/` (retrieved: Feb'09), Feburary 2009.

[SREZ05] J. Sivic, B. Russell, A. Efros, and A. Zisserman. Discovering Objects and their Location in Images. In *Proc. Int. Conf. Computer Vision*, pages 370–377, October 2005.

[SS01] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001.

[SS07] I. Simon and S. Seitz. A Probabilistic Model for Object Recognition, Segmentation, and Non-Rigid Correspondence. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–7, June 2007.

[SSTK08] Y. Sun, S. Shimada, Y. Taniguchi, and A. Kojima. A Novel Region-based Approach to Visual Concept Modeling using Web Images. In *Int. Conf. Multimedia*, pages 635–638, October 2008.

[ST94] J. Shi and C. Tomasi. Good Features to Track. In *Proc. Int. Conf. on Pattern Recognition*, pages 593–600, June 1994.

[SvZ08] B. Sigurbjörnsson and R. van Zwol. Flickr Tag Recommendation based on Collective Knowledge. In *Proc. Int. Conf. on World Wide Web*, pages 327–336, 2008.

[SW05a] C. Snoek and M. Worring. Multimedia Event-Based Video Indexing using Time Intervals. *IEEE Trans. Multimedia*, 7(4):638–647, 2005.

[SW05b] C. Snoek and M. Worring. Multimodal Video Indexing: A Review of the State-of-the-art. *Multimedia Tools and Applications*, 25(1):5–35, 2005.

[SWdR+08] C. Snoek, M. Worring, O. de Rooij, K. van de Sande, R. Yan, and A. Hauptmann. VideOlympics: Real-Time Evaluation of Multimedia Retrieval Systems. *IEEE MultiMedia*, 15(1):86–91, 2008.

[SWG+05] C. Snoek, M. Worring, J.-M. Geusebroek, D. Koelma, and F. Seinstra. On the Surplus Value of Semantic Video Analysis Beyond the Key Frame. In *Proc. IEEE Int. Conf. on Multimedia & Expo*, page 4pp., July 2005.

[SWG+06] C. Snoek, M. Worring, J.-M. Geusebroek, D. Koelma, F. Seinstra, and A. Smeulders. The Semantic Pathfinder: using an Authoring Metaphor for Generic Multimedia Indexing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(10):1678–1689, 2006.

[SWSJ00] A. Smeulders, M. Worring, S. Santini, and A. Gupta R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE*

Trans. Pattern Analysis and Machine Intelligence, 22(12):1349–1380, 2000.

[SWvG+06] C. Snoek, M. Worring, J. van Gemert, J. Geusebroek, and A. Smeulders. The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia. In *Proc. Int. Conf. on Multimedia*, pages 225–226, October 2006.

[SZ03] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proc. Int. Conf. Computer Vision*, pages 1470–1477, October 2003.

[SZ04] J. Sivic and A. Zisserman. Video Data Mining using Configurations of Viewpoint Invariant Regions. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 488–495, June 2004.

[SZ06] J. Sivic and A. Zisserman. Video Google: Efficient Visual Search of Videos. In *Toward Category-Level Object Recognition*, pages 127–144. Springer-Verlag New York, Inc., 2006.

[SZB08] G. Schindler, L. Zitnick, and M. Brown. Internet Video Category Recognition. In *Proc. First Internet Vision Workshop*, pages 1–7, June 2008.

[SZTS06] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Background Cut. In *Proc. Europ. Conf. Computer Vision*, pages 628–641, May 2006.

[TA06] S. Todorovic and N. Ahuja. Extracting Subimages of an Unknown Category from a Set of Images. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 927–934, June 2006.

[Tek95] A. Tekalp. *Digital Video Processing*. Prentice-Hall, Inc., 1995.

[TK91] C. Tomasi and T. Kanade. Detection and Tracking of Point Features. Technical report, Carnegie Mellon University, 1991.

[TL07] J. Tang and P. Lewis. A Study of Quality Issues for Image Auto-Annotation with the Corel Dataset. *IEEE Trans. Circuits and Systems for Video Technology*, 17(3):384–389, 2007.

[TM93] P. Torr and D. Murray. Outlier Detection and Motion Segmentation. In *Proc. SPIE Sensor Fusion Conf.*, pages 432–443, September 1993.

[TMY78] H. Tamura, S. Mori, and T. Yamawaki. Textural Features Corresponding to Visual Perception. *IEEE Trans. System, Man, Cybernetics*, 8(6):460–472, 1978.

[TNS07] J. Tesic, A. Natsev, and J. Smith. Cluster-based Data Modeling for Semantic Video Search. In *Proc. Int. Conf. Image and Video Retrieval*, pages 595–602, July 2007.

[Tor03] A. Torralba. Contextual Priming for Object Detection. *Int. J. Comput. Vision*, 53(2):169–191, 2003.

[Tou02] A. Tourapis. Enhanced Predictive Zonal Search for Single and Multiple Frame Motion Estimation. In *Proc. SPIE Conf. Visual Communications and Image Processing*, pages 1069–1079, September 2002.

[Tur93] B. Turlach. Bandwidth Selection in Kernel Density Estimation: A Review. In *CORE and Institut de Statistique*, pages 23–49, 1993.

[TV07] R. Tron and R. Vidal. A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[TZ00] P. Torr and A. Zisserman. Feature Based Methods for Structure and Motion Estimation. In *Proc. Int. Workshop on Vision Algorithms*, pages 278–294, September 2000.

[UB08] A. Ulges and T. Breuel. Segmentation by Combining Optical Flow with a Color Model. In *Proc. Int. Conf. on Pattern Recognition*, December 2008.

[UBB09] A. Ulges, D. Borth, and T. Breuel. Visual Concept Learning from Weakly Labeled Web Videos (submitted for publication). In *Video Search and Mining*. Springer-Verlag, 2009.

[UKSB08] A. Ulges, M. Koch, C. Schulze, and T. Breuel. Learning TRECVID'08 High-level Features from YouTube^{TM}. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2008.

[Ulg07] A. Ulges. Motion Interpretation using Adaptive Search of Transformation Space. Technical report, Department of Computer Science, University of Kaiserslautern, 2007.

[ULKB07] A. Ulges, C. Lampert, D. Keysers, and T. Breuel. Optimal Dominant Motion Estimation using Adaptive Search of Transformation Space. In *Proc. DAGM-Symposium*, pages 204–213, September 2007.

[USA06] YouTube Serves up 100 Million Videos a Day Online. in USA Today (Garnnett Company, Inc.); available from `http://www.usatoday.com/tech/news/2006-07-16-youtube-views_x.htm` (retrieved: Sep'08), July 2006.

[USB08] A. Ulges, C. Schulze, and T. Breuel. Multiple Instance Learning on Weakly Labeled Videos. In *Proc. SAMT Workshop on Cross-Media Information Analysis and Retrieval*, December 2008.

[USKB07] A. Ulges, C. Schulze, D. Keysers, and T. Breuel. Content-Based Video Tagging for Online Video Portals. In *Proc. MUSCLE/ImageCLEF Workshop*, September 2007.

[USKB08a] A. Ulges, C. Schulze, D. Keysers, and T. Breuel. A System that Learns to Tag Videos by Watching Youtube. In *Proc. Int. Conf. on Vision Systems*, pages 415–424, May 2008.

[USKB08b] A. Ulges, C. Schulze, D. Keysers, and T. Breuel. Identifying Relevant Frames in Weakly Labeled Videos for Training Concept Detectors. In *Proc. Int. Conf. Image and Video Retrieval*, pages 9–16, July 2008.

[USKB09] A. Ulges, C. Schulze, M. Koch, and T. Breuel. The Challenge of Tagging Online Video. *Comp. Vis. Img. Underst. (submitted for publication)*, 2009.

[UVNS02] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual Features of Intermediate Complexity and their Use in Classification. *Nature Neuroscience*, 5(7):682–687, 2002.

[vdSGS08] K. van de Sande, T. Gevers, and C. Snoek. A Comparison of Color Features for Visual Concept Classification. In *Proc. Int. Conf. Image and Video Retrieval*, pages 141–150, July 2008.

[vES00] M. van. Erp and L. Schomaker. Variants of the Borda Count Method for Combining Ranked Classifier Hypotheses. In *Proc. Int. Workshop Frontiers in Handwriting Recognition*, pages 443–452, September 2000.

[vGV+06] J. van Gemert, J. Geusebroek, C. Veenman, C. Snoek, and A. Smeulders. Robust Scene Categorization by Learning Image Statistics in Context. In *CVPR Workshop on Semantic Learning Applications in Multimedia*, June 2006.

[VGWK08] T. Vaudrey, D. Gruber, A. Wedel, and J. Klappstein. Space-Time Multi-Resolution Banded Graph-Cut for Fast Segmentation. In *Proc. DAGM-Symposium*, pages 203–213, June 2008.

[VID08] 1st Annual Video Search Summit. available from `http://www.videosearchnews.com/` (retrieved: Feb'09), April 2008.

[VIP09] Evolution Robotics VIPR Technology. available from `http://www.evolution.com/core/ViPR/` (retrieved: Feb'09), February 2009.

[WA93] J. Wang and E. Adelson. Layered Representation for Motion Analysis. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 361–366, June 1993.

[WAB03] J. Wills, S. Agarwal, and S. Belongie. What Went Where. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 37–44, June 2003.

[WCGH99] H. Wactlar, M. Christel, Y. Gong, and A. Hauptmann. Lessons Learned from Building a Terabyte Digital Video Library. *Computer*, 32(2):66–73, 1999.

[Wei97] Y. Weiss. Smoothness in Layers: Motion Segmentation using Nonparametric Mixture Estimation. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 520–527, June 1997.

[WHS+06] M. Wang, X.-S. Hua, Y. Song, X. Yuan, S. Li, and H.-J. Zhang. Automatic Video Annotation by Semi-supervised Learning with Kernel Density Estimation. In *Proc. Int. Conf. on Multimedia*, pages 967–976, October 2006.

[Wir09] O. Wirjadi. A Branch and Bound Algorithm for Finding the Modes in Kernel Density Estimates. *Int. J. Computational Intelligence and Applications*, 8(1):17–35, 2009.

[WJ05] J. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *Int. Conf. Computer Vision*, pages 756–763, October 2005.

[WLL+07] D. Wang, X. Liu, L. Luo, J. Li, and B. Zhang. Video Diver: Generic Video Indexing with Diverse Features. In *Proc. Int. Workshop Multimedia Information Retrieval*, pages 61–70, September 2007.

[WLT08] W. Wang, Y. Luo, and G. Tang. Object Retrieval using Configurations of Salient Regions. In *Proc. Int. Conf. Image and Video Retrieval*, pages 67–74, July 2008.

[WLW04] T.-F. Wu, C.-J. Lin, and R. Weng. Probability Estimates for Multiclass Classification by Pairwise Coupling. *J. Mach. Learn. Res.*, 5:975–1005, 2004.

[WS06] J. Winn and J. Shotton. The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 37–44, June 2006.

[WS08] K. Wnuk and S. Soatto. Filtering Internet Image Search Results Towards Keyword Based Category Recognition. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[WXZG07] J. Wang, W. Xu, S. Zhu, and Y. Gong. Efficient Video Object Segmentation by Graph-Cut. In *Proc. Int. Conf. on Multimedia and Expo*, pages 496–499, July 2007.

[WZLM08] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma. Annotating Images by Mining Image Search Results. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(11):1919–1932, 2008.

[YB05] K. Yanai and K. Barnard. Probabilistic Web Image Gathering. In *Int. Workshop on Multimedia Inf. Retrieval*, pages 57–64, November 2005.

[YCKH07] A. Yanagawa, S.-F. Chang, L. Kennedy, and W. Hsu. Columbia University's Baseline Detectors for 374 LSCOM Semantic Visual Concepts. Technical report, Columbia University, 2007.

[YDF05] C. Yang, M. Dong, and F. Fotouhi. Region Based Image Annotation through Multiple-instance Learning. In *Proc. Int. Conf. on Multimedia*, pages 435–438, November 2005.

[YH08] J. Yang and A. Hauptmann. (Un)Reliability of Video Concept Detection. In *Proc. Int. Conf. Image and Video Retrieval*, pages 85–94, July 2008.

[YHC07] A. Yanagawa, W. Hsu, and S.-F. Chang. Brief Descriptions of Visual Features for Baseline TRECVID Concept Detectors. Technical report, Columbia University, 2007.

[YKA02] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting Faces in Images: a Survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[YLP00] C. Yang and T. Lozano-Perez. Image Database Retrieval with Multiple-Instance Learning Techniques. In *Proc. Int. Conf. on Data Engineering*, pages 223–243, February 2000.

[YMH+07] B. Yang, T. Mei, X. Hua, L. Yang, S. Yang, and M. Li. Online Video Recommendation Based on Multimodal Fusion and Relevance Feedback. In *Proc. Int. Conf. Image and Video Retrieval*, pages 73–80, July 2007.

[YOU] "Youtube". in Wikipedia: The Free Encyclopedia; (Wikimedia Foundation Inc.) [encyclopedia on-line]; available from `http://en.wikipedia.org/wiki/YouTube` (retrieved: Sep'08).

[YOU06] Google closes Acquisition of YouTube. YouTube Press Release; available from `http://www.youtube.com/press_room_entry?entry=AwPf9c9qJDc` (retrieved: Feb'09), November 2006.

[YSR05] A. Yavlinsky, E. Schofield, and S. Rüger. Automated Image Annotation using Global Features and Robust Nonparametric Density Estimation. In *Proc. Int. Conf. Image and Video Retrieval*, pages 507–517, July 2005.

[Yua07] Yuan, J. et al. THU and ICRC at TRECVID 2007. In *Proc. TRECVID Workshop (unreviewed workshop paper)*, November 2007.

[YWX+07] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang. A Formal Study of Shot Boundary Detection. *IEEE Trans. Circuits and Systems for Video Technology*, 17(2):168–186, 2007.

[YYH07] J. Yang, R. Yan, and A. Hauptmann. Cross-Domain Video Concept Detection using Adaptive SVMs. In *Proc. Int. Conf. on Multimedia*, pages 188–197, September 2007.

[ZCPR03] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Comput. Surv.*, 35(4):399–458, 2003.

[Zhu05] X. Zhu. Semi-supervised Learning Literature Survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[ZL01] D. Zhang and G. Lu. Segmentation of Moving Objects in Image Sequence: A Review. *Circuits, Systems and Signal Processing*, 20(2):143–183, 2001.

[ZMLS07] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *Int. J. Comput. Vis.*, 73(2):213–238, 2007.

[ZMZP08] L. Zelnik-Manor, S. Zanetti, and P. Perona. A Walk Through the Web's Video Clips. In *Proc. First Internet Vision Workshop*, pages 1–7, June 2008.

[ZT06] J. Zobel and T.Hoad. Detection of Video Sequences using Compact Signatures. *ACM Trans. Inf. Syst.*, 24(1):1–50, 2006.

Curriculum Vitae

Name:	Adrian Ulges
Date of Birth:	February 17, 1980
Place of Birth:	Koblenz / Germany

Education

School Education	Goethe-Gymnasium Bad Ems, *A*bitur, 1999
University Education:	University of Kaiserslautern and University of Koblenz, Germany *Diploma in Computer Science (University of Kaiserslautern)*, 2005

Academic and Professional Experience

Oct 2003 – Sep 2008	Fellow of PhD program of computer science at the University of Kaiserslautern
Aug 2005 – Nov 2005	Internship with Google Inc, Mountain View, CA
Oct 2008 – present	Researcher with DFKI GmbH, Kaiserslautern, Germany