FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Intelligent Video Annotation and Retrieval Techniques

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der technischen Wissenschaften

by

Robert Sorschag
Registration Number 0060423

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: a.Univ.-Prof., Dr. Horst Eidenberger

The dissertation has been reviewed by:

(a.Univ.-Prof., Dr. Horst Eidenberger)

(Juniorprof. Dr. habil. Ansgar Scherp)

Wien, 17.09.2012

(Robert Sorschag)

Erklärung zur Verfassung der Arbeit

Robert Sorschag
Esterhazygasse 12/1/19, 1060 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Robert Sorschag)

Acknowledgements

First and foremost I want to thank my family and friends for supporting me in every possible way during my dissertation project. I further thank my office mates Dalibor Mitrović and Matthias Zeppelzauer for many fruitful discussions about thesis-related topics and thesis-unrelated topics. Next, I want to express my gratitude to my supervisors Associate Professor Horst Eidenberger and Juniorprofessor Ansgar Sherp.

I am grateful to all my colleagues who made working fun and special thanks go to Ingrid Lisa for making my live easier when dealing with administration. Additionally, I want to thank Werner Bailer, Georg Thallinger and all the other guys from Joanneum Research who participated jointly with me at various TRECVID challenges. Finally, I thank the Austrian Research Program FIT-IT for funding the dissertation fellowship project IV-ART that leads to this dissertation.

Robert Sorschag

Abstract

Videos are an integral part of current information technologies and the web. The demand for efficient retrieval rises with the increasing number of videos, and thus better annotation tools are needed as today's retrieval systems mainly rely on manually generated metadata. The situation is even more critical when it comes to user-generated videos where rough and inaccurate annotations are the common practice. Attempts to employ content-based analysis for video annotation and retrieval already exist, but they are still in an infant stage compared to the retrieval of web documents.

In this work, we address the use of object recognition techniques to annotate *what* is shown *where* in videos. These annotations are suitable to retrieve specific video scenes for object related text queries, thought the manual generation of such metadata would be impractical and expensive. A sophisticated presentation of the retrieval results is further exploited that indicates the relevance of the retrieved scenes at a first glance. The presented semi-automatic annotation approach can be used in an easy and comfortable way, and it builds on a novel framework with following outstanding features. First, it can be easily integrated into existing video environments. Second, it is not based on a fixed analysis chain but on an extensive recognition infrastructure that can be used with all kinds of visual features, matching and machine learning techniques. New recognition approaches can be integrated into this infrastructure with low development costs and a configuration of the used recognition approaches can be performed even on a running system. Thus, this framework might also benefit from future advances in computer vision. Third, we present an automatic selection approach to support the use of different recognition strategies for the annotation of different objects. Moreover, visual analysis can be performed efficiently on distributed, multi-processor environments and the resulting video annotations and low-level features can be stored in a compact form.

We demonstrate the proposed annotation approach in an extensive case study with promising results. A video object annotation prototype as well as the generated scene classification groundtruth are freely available to foster reproducible research. Additional contributions of this work consider the generation of motion-based and segmentation-based features and their use for specific annotation tasks, such as the detection of action scenes in professional and user-generated video. Furthermore, we participated at the two tasks instance search and semantic indexing of the TRECVID challenge in the three consecutive years 2010, 2011, and 2012.

Kurzfassung

Videos sind ein wesentlicher Bestandteil moderner Informationssysteme und des Webs. Seit der Einführung der ersten Videoportale Mitte des letzten Jahrzehnts gibt es ein stetiges Wachstum der verfügbaren Videos und damit einhergehend die Notwendigkeit effizienterer Videosuche. Aktuelle Suchsysteme arbeiten hauptsächlich auf manuell erzeugten Metadaten, die den Nachteil haben, dass sie Videoinhalte oft nur grob und ungenau beschreiben. Deswegen sollen Videoannotationssysteme, die auf inhaltsbasierte Analyse setzen, Abhilfe schaffen und die Videosuche auf ein ähnliches Niveau bringen, wie man es heute von der Online-Suche nach Textdokumenten und Webseiten gewohnt ist.

Die vorliegende Dissertation beschäftigt sich mit der Verwendung automatischer Objekterkennung für die Annotation von Personen, Objekten und Orten. Nach der Beschlagwortung können Videoszenen dieser Objekte mit Google-ähnlichen Suchanfragen gefunden werden. Durch eine ausgeklügelte Präsentation der gefundenen Videoszenen wird die Relevanz einzelner Suchresultate sofort sichtbar. Die vorgestellten Annoationstechniken basieren auf einem neuen Objekterkennungs-Framework, das in verschiedenste Videoumgebungen eingebunden werden kann und Objekterkennung mit einer flexiblen Verwendung von visuellen Features, Vergleichsalgorithmen und Techniken des maschinellen Lernens ermöglicht. Neue Methoden können mit geringem Entwicklungsaufwand in dieses Framework integriert werden. Dies erlaubt eine schnelle Verwendung neuer Entwicklungen und kann deswegen speziell für zukünftige Forschungen einen wichtigen Beitrag leisten. Desweiteren bietet das Framework eine automatische Konfigurationsauswahl die es möglich macht verschiedene Algorithmen für die Annotation von verschiedenen Objekten zu verwenden. Die Unterstützung verteilter Computersysteme und das kompakte Speichern der erzeugten Daten gewährleisten außerdem hohe Effizienz.

Im Laufe des Projektes wurde mit den vorgestellten Techniken ein Videoannotations-Prototyp entwickelt um eine umfassende Fallstudie durchzuführen. Dieser Prototyp ist ebenso wie die verwendeten Videodaten und einige der resultierenden Publikationen öffentlich verfügbar. Weitere wissenschaftliche Beiträge der Dissertation behandeln bewegungsbasierte und segmentationsbasierte Features, welche für spezielle Einsatzgebiete wie die automatische Actionszenenerkennung geeignet sind. Zwischen 2010 und 2012 haben wir darüberhinaus bei TRECVID, dem größten internationalen Wettbewerb für inhaltsbasierte Videosuche, teilgenommen und dabei vielversprechende Ergebnisse erzielt.

Contents

CHAPTER 1

Introduction

Intelligent video annotation and retrieval techniques are required to use video more effectively. This PhD thesis was performed in the context of a dissertation fellowship project that aims at the development of such annotation and retrieval techniques. In the following, the need for improved techniques is first motivated before the contributions and an overview of this thesis are given.

1.1 Motivation

Video streaming has become one of the most important services on the web and it accounts for a large share of today's internet traffic [50]. Since YouTube went online in 2005 an immense number of videos made their way through the internet and more than 50% of all web users visit online video portals on a daily basis [6]. Despite this success, the retrieval capabilities of popular portals are somehow premature as it is still more difficult to retrieve videos on the web than to retrieve websites and text documents. Another sign of this prematurity is that only a small partition of all video views originates from a search-oriented retrieval whereas link-based views dominate. Such links stem from recommended, top-viewed, and related video entries inside a portal as well as from postings in social networks and video recommendations in e-mails.

Search-based retrieval usually operates on textual metadata that is generated during video annotation. We distinguish between the retrieval of user-generated and professional video. Much less professional content exist and these videos are accompanied by richer metadata, such as sub-titles and a list of involved actors. In contrast, the retrieval of user-generated videos mainly operates on the title and it only succeeds if the user knows this title accurately, if she searches for very popular videos, or for videos that contain some outstanding features. Object and scene retrieval is generally out of reach of current video portals, thought it is very important for certain video archives. Broadcast channels, for instance, need to maintain their archives in a way that they gain access to archived material of specific dates, events, or persons in order to reuse these scenes in present shows.

Content-based analysis techniques are rarely used to annotate videos, and content-based queries are even less popular for video retrieval because it is difficult to obtain appropriate query content in the first place. Moreover, the meaning of such queries is often ambiguous and it is not always clear what the interesting part of an example is. Thus, more detailed and better annotations are the key to improve video retrieval, and more powerful tools and automation techniques are required to generate such annotations as the manual metadata generation is a time consuming, tedious, and overall expensive task. On the other hand, users need to be motivated to annotate their videos accurately.

Recently, automated approaches for individual object and object class recognition achieved promising results on realistic benchmarks [85] and they provide a good opportunity to improve video annotation and retrieval. Thus, the main research question of this thesis is: *How to employ object recognition for intelligent video annotation and retrieval tools?* A couple of points have to be considered to answer this question. First, many object recognition approaches already exist and it requires experience and a fairly advanced level of computer vision skills to select an appropriate approach for the recognition of specific objects. This is especially true as different approaches are best suited to recognize different objects [97]. The choice of visual features depends on the attributes of the objects and their discriminative factors (color, texture, shape, motion), on the one hand, and the video material, on the other hand. Furthermore, recent studies [16,288,342] show that the performance of most recognition approaches can be significantly improved when they are adapted to certain tasks, domains, or datasets. In addition, [238] discovered that the currently used visual features seem to provide the main reason why the human visual system outperforms existing computer vision approaches in recognition tasks. Therefore, it is very probable that new and more powerful features will be developed in the future.

Multimedia analysis is an active research topic since several decades and some analysis tools already achieved marketability. It is important to learn from existing systems, to build on their results, and to integrate available solutions. A high degree of flexibility is required to achieve this and additional tools are needed to develop and integrate object recognition approaches into video annotation and retrieval systems. It is further important to support reproducible research by the use of open datasets that capture real-world retrieval tasks realistically, not only for professional content but also for user-generated content with low quality and very long shots. Finally, the common practice how video retrieval results are presented to the users is suboptimal as only one or a few keyframes are independently chosen for the actual query [315]. Users often have to view large parts of a video in this approach to verify if a retrieval result fits to the submitted query or not. Thus, query-dependent keyframes that show the interesting content at a first glance are considered as improvement.

We stick to these ideas and try to narrow the gap between video retrieval and web document retrieval from several sides. First, we want to outline the current state-of-the-art, open points, and the available room for improvements in the areas of video annotation and retrieval, and object recognition. New visual features are then proposed that are suited for specific retrieval tasks and that combine low-level features to semantically higher features in order to describe objects of complex real-world scenes. We propose an extendible object recognition infrastructure as well as a flexible object annotation approach that represents an interface between video portals and computer vision research. Resulting annotations can lead to a more enjoyable video retrieval experience with a higher percentage of satisfying retrieval results because they describe those parts of a video that are usually not mentioned by manually generated video annotations. Objects

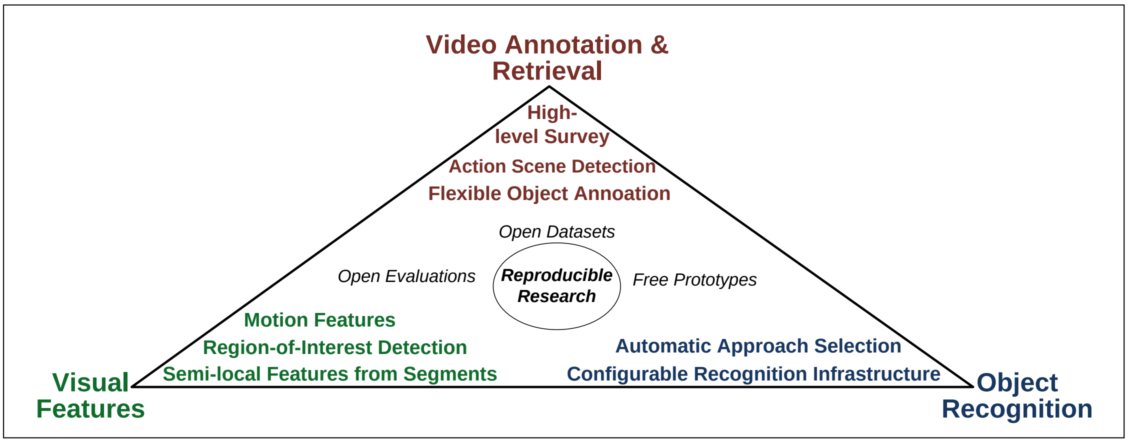

Video Annotation & Retrieval
High-level Survey
Action Scene Detection
Flexible Object Annoation
Open Datasets
Open Evaluations
Reproducible Research
Free Prototypes
Motion Features
Region-of-Interest Detection
Semi-local Features from Segments
Automatic Approach Selection
Configurable Recognition Infrastructure
Visual Features
Object Recognition

Figure 1.1: Overview of this thesis. Various contributions to the three research directions video annotation and retrieval, object recognition, and visual features are made with an emphasis on reproducibility.

that have been annotated in one video might be automatically annotated in other videos as well. In addition to improved video annotation and retrieval capabilities, the outcome of this thesis might help to detect unwanted videos that show inappropriate content or copyright protected content.

1.2 Contributions

The contributions of this thesis are manifold and range from the presentation of novel visual features over an extensive recognition infrastructure to specific techniques for video annotation and retrieval, see Figure 1.1. First, a high-level survey of current video annotation and retrieval systems draws the big picture that is common for all of today's systems, their distinguishing features, and that points out research trends and access points for further reading. This survey contributes to multimedia research with a proper overview of the field whereas all existing studies are restricted to specific aspects, for instance on object-based video retrieval and the underlying computer vision techniques [337]. In addition to specific techniques for action scene detection and semantic concept annotation, we propose a flexible object annotation approach for online video portals that provides an iterative data collection where only one object example has to be annotated by the user in the first place. More instances are suggested by the approach then, and it even provides the possibility of a fully automatied object annotation. The resulting annotations enable video retrieval on an object and scene level for professional and user-generated content. Moreover, it enables a sophisticated presentation of the retrieval results that indicates the relevance of the retrieved scenes at a first glance instead of always using the same keyframes independent of the submitted queries. We further developed a prototype to demonstrate the annotation capabilities and to perform an extensive case study.

A configurable object recognition infrastructure was first developed that significantly facilitates the generation of compute vision prototypes and products. This infrastructure can be extended by new features and recognition approaches as well as existing libraries. Simple configurations are thereby sufficient to adjust the used recognition approaches even on a running system, and efficient analysis is further achieved by the support of distributed multi-processor environments. Furthermore, a technique to select and customize recognition approaches for specific tasks, domains, and datasets is proposed that operates on top of this recognition infrastructure. In contrast to existing works, this approach tries to automate the simultaneous selection and customization of the entire recognition process with all kinds of visual features, matching and machine learning strategies. It allows non-experts to utilize object recognition for their applications and enables experts to develop and evaluate new recognition approaches.

We further developed a set of novel visual features that are especially suited for video annotation and retrieval. On the one hand, we present two motion features for action scene annotation with an implicit presentation of the underlying video structure. On the other hand, we propose semi-local features that exploit object segmentation as a pre-processing step for class-level object recognition. The term semi-local features indicates that the proposed features are extracted from interest regions that contain entire objects instead of arbitrary object parts. In particular, we investigate the impact of feature generation approaches from differently transformed object regions. In contrast to this approach, class-level object recognition is usually tackled with global or local image features. Moreover, an automatic object region detector is given to generate input regions for these semi-local features in the context of video object annotation.

Finally, this thesis contributes to reproducible research in multiple ways. The experiments are done on open datasets where possible, and otherwise we published the used datasets and annotations. Every proposed technology was individually evaluated and we repeatedly participated at the TRECVID video retrieval benchmark with the Institute for Information and Communication Technologies of the applied research company Joanneum Research [147]. Thereby, we worked on the two tasks *semantic indexing* and *instance search*, and helped to annotate a large corpus of videos that are used for the evaluations. Moreover, a video object annotation prototype is freely available to the public in addition to some of the resulting publications (see next section) that are published as open access variant.

1.3 Resulting Publications

Journals and Book Chapters

- Sorschag R.: *A Flexible Object-of-Interest Annotation Framework for Online Video Portals.* Future Internet (ISSN 1999-5903), special issue 'Visual Information Retrieval'. 2012.

- Sorschag R.: *Object Detection with Semi-local Features.* In print: Advances in Intelligent and Soft Computing, Springer-Verlag. 2012.

- Sorschag R.: *A High-Level Survey of Video Annotation and Retrieval Systems.* In print: International Journal of Multimedia Technology. 2012.

Conferences

- Sorschag R.: *Semi-Local Features for the Classification of Segmented Objects*. In: 1st International Conference on Pattern Recognition Applications and Methods (ICPRAM). Vilamoura, Algarve, Portugal, 6-8 February, 2012.

- Sorschag R.: *How to Select and Customize Object Recognition Approaches for an Application?* In: The 18th International Conference on MultiMedia Modeling (MMM). Klagenfurt, Austria, January 4-6, 2012.

- Sorschag R.: *CORI: A Configurable Object Recognition Infrastructure*. In: IEEE International Conference on Signal & Image Processing Applications 2011 (ICSIPA). Kuala Lumpur, Malaysia, 16-18 November, 2011.

- Sorschag R. and Hoerhan M.: *Action Scene Detection from Motion and Events*. In: IEEE International Conference on Image Processing (ICIP). Brussels, Belgium, 11-14 September, 2011.

- Sorschag R., Moerzinger R. and Thallinger G.: *Automatic Region of Interest Detection in Tagged Images*. In: IEEE International Conference on Multimedia and Expo (ICME). New York, USA, June 28 - July 3, 2009.

Workshops

- Bailer W., Sorschag R., Moezinger R., Hoelzl S., Lee F. and Stiegler H.: *JOANNEUM RESEARCH and Vienna University of Technology at TRECVID 2011*. In: Proceedings of TRECVID Workshop. Gaithersburg, MD, USA 2011.

- Bailer W., Sorschag R., Lee F., Stiegler H. and Schwendt G.: *JOANNEUM RESEARCH and Vienna University of Technology at TRECVID 2010*. In: Proceedings of TRECVID Workshop. Gaithersburg, MD, USA 2010.

1.4 Summary by Section

Section 2 starts with a broad overview of current video annotation and retrieval techniques. The focus lies on the purpose and functionality of existing systems and the requirements and intentions of their users. Object recognition approaches including visual features, matching, and classification strategies are then explained because they form the basis of the video annotation and retrieval technologies that are presented in this thesis. Finally, we review dataset issues, the popular evaluation methods and initiatives, as well as tools that facilitate the development of annotation and recognition systems. Research trends and access points for further reading are also mentioned that particularly address researchers which are new to the field.

The configurable object recognition infrastructure CORI is presented in Section 3. This infrastructure can be adapted to the needs of a broad spectrum of tasks without writing a single line of code. Instead, it can be configured to generate various visual features and to perform training and recognition. The configuration format is simple and tools to generate these configurations make it easy to use, even for developers with little computer vision experience. New algorithms and approaches can be added in a reusable way and further tools are provided to cover all parts of object recognition applications from data acquisition over execution and storage to the final result presentation. Moreover, performance considerations have been taken into account by the reuse of intermediate results and the support of multi-processor architectures. After this, we propose the simultaneous selection and customization of the entire recognition process from an annotated set of sample images or videos and precisely specified task requirements. This approach uses CORI in combination with iterative analysis strategies to be practicable for real-world applications in the area of video annotation. All proposed feature extraction, video annotation and retrieval techniques of this thesis build upon CORI and the automatic approach selection.

Section 4 first presents semi-local features that exploit object segmentation as a preprocessing step for object recognition. The term semi-local features indicates that the proposed features are locally extracted from the image but globally extracted from the object. In particular, we investigate the impact of feature generation approaches from differently transformed object regions. These transformations are, on the one hand, done with several object-background modifications and bounding-boxes. On the other hand, state-of-the-art texture and color features as well as different dissimilarity measures are compared against each other. We then describe a local feature matching-based object region detector to segment repeatedly shown objects. This detector works object independent on nearby video frames or similarly annotated images. Finally, gist-based global motion features and SIFT-based local motion features are presented that are especially designed for action scene classification.

Section 5 addresses the use of object recognition techniques to annotate *what* is shown *where* in video collections. These annotations are suitable to retrieve specific video scenes for object related text queries. The proposed approach possesses some outstanding features that offer good prospects for its application in various video annotation and retrieval systems. It can be easily integrated into different video environments and it is not based on a fixed analysis chain, which means that future advances in computer vision can be applied as soon as they are available. We further present an automatic selection approach to support the use of different recognition strategies for different objects. A database schema is then given to store the resulting video annotations as well as the off-line generated low-level features in a compact form. Moreover, we present a prototype that demonstrates the object annotation capabilities of this approach and that is used for an extensive case study.

In Section 6, we describe the used annotation and retrieval strategies of our TRECVID participations in the two tasks, semantic indexing and instance search. Moreover, we present a scene classification technique to separate action scenes from non-action scenes. In contrast to existing work, the proposed system does not consider the shot structure of video. Overall, these evaluations have lead to promising results and we draw conclusions in Section 7. Open topics and an outlook of future research perspectives are also given in this section.

CHAPTER 2

Video Annotation and Retrieval

Content-based analysis methods for video annotation and retrieval are proposed from several research directions. In the following, we review this research with an emphasis on scene-based and object-based methods. Furthermore, we discuss the state-of-the-art of object recognition, automatic approach selection, and recognition infrastructures. These topics are directly related to the work of this dissertation thesis.

2.1 Visual Content Analysis

Videos are an integral part of current information technologies and the web. The demand for efficient retrieval rises with the increasing number of videos, which is equally true for video annotation techniques as matadata is the primary source of most retrieval systems. In this section, we start with a general definition of video documents before different aspects of annotation and retrieval systems are discussed in combination with user intentions and interaction processes.

2.1.1 Content and Metadata

The classical view of video documents is characterized by two well-defined role models for video content generation and usage. On the one hand, users consume, retrieve, and share specific videos. Metadata plays a central role in this context to shorten the bridge between users and videos, as it facilitates efficient retrieval [198]. On the other hand, producers generate videos and annotate them with metadata using specific tools. However, this view changes when it comes to web videos where professional content and user-generated content have to be distinguished. In addition to an alternated role model, user-generated videos show a significant shorter content production scale than professional videos. Thus, IMDB [140], the world's largest online database of professional movies, carries about 1 million titles that are produced since 1888 while YouTube [359] counts above 65000 new uploads every day [47].

Generally, video documents consist of *audio-visual content* and *metadata*. A wide variety of video types exist, starting from traditional content (movies, sport, music, and news) over

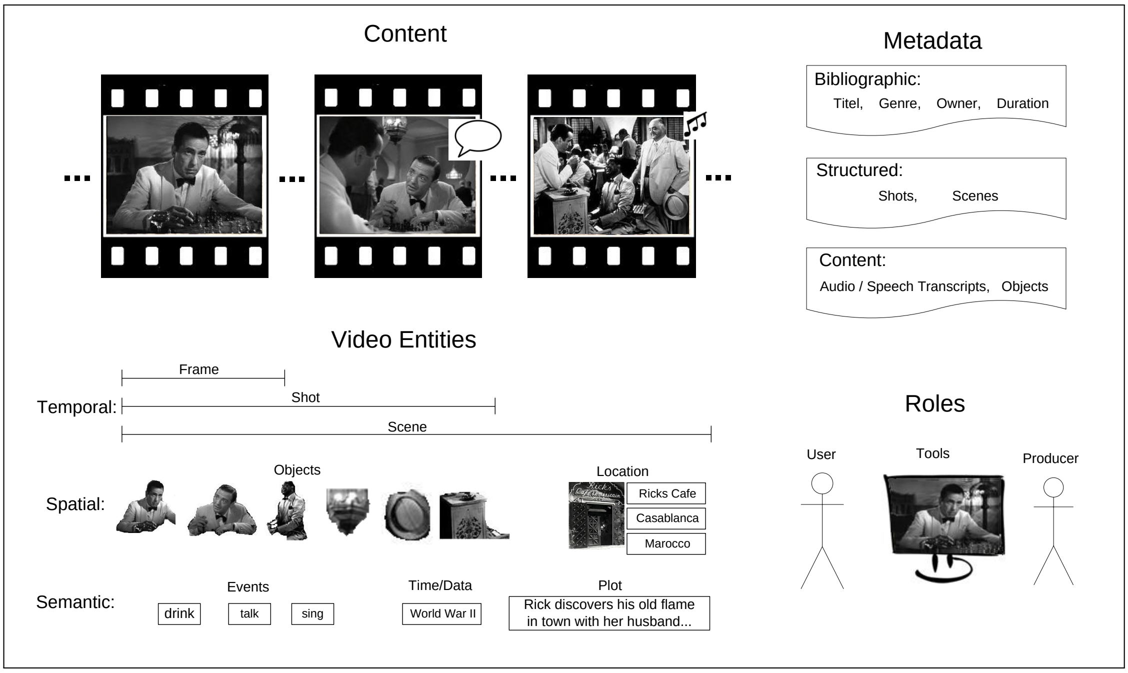


Figure 2.1: Overview of video documents using the classical movie Casablanca. Audio-visual content is accompanied by metadata that can be distinguished into bibliographic, structured, and content metadata. The video entities are similarly divided into temporal, spatial, and semantic entities.

personal content (home videos) to content that is meant to express thoughts and opinions or to share expertise (video blogs and tutorial videos) [285]. Consequently, videos answer the purpose of entertainment, information, communication, and data analysis [125]. As shown in Figure 2.1, all kinds of video are structured in a hierarchy of frames, shots, and scenes on the *temporal level*. Moreover, objects and locations as well as events and background information are the *video entities* on the *spatial* and *semantic level*, respectively [286].

In accordance to these video entities, [262] distinguishes between three types of metadata. *Bibliographic metadata* describes general video attributes, such as the video title, its genre, owner, duration, and the number of views on a video portal. *Structural metadata* includes the hierarchy of scenes and shots, while *content metadata* describes information like the speech and audio transcripts of a video. Ideally, such metadata should be accurate, complete, and cost-effective in its generation [198], but this is not the current practice. However, metadata for professional content is often quite extensive compared to user-generated content, as it includes information about the cast and crew, the story line, awards, box office results, and even multilingual subtitles are often given. In contrast, most user-generated videos are only annotated with a title, a short description, some tags, and user ratings.

Beside video annotation and retrieval systems that are extensively discussed in this section, many tools exist for the video production process to perform different tasks, such as camera stabilization, video restoration, and post-processing. Moreover, editing software is used to cut and join film sequences into a continuous video. In the context of user-generated video, production and editing tools are less important and less sophisticated. Interoperability between these tools is nevertheless important and the following aspects have to be considered: A variety of video and audio codecs exist that use different compression techniques. The generated metadata can be stored in different structures, the same information can be given using different terms (synonyms), and it might be unclear what a specific term describes at all (inherent semantics) [40]. Thus, several de-jure and de-facto standards exist for multimedia content, its metadata, and for retrieval techniques in order to harmonize the exchange formats of different tools. Table 2.1 shows common standards that are either introduced by the industry or from initiatives for standardization and research.

2.1.2 Video Annotation

As noted in [218], it is generally more difficult to generate video annotations than to add notes to text-based materials, such as paper or text files. Specific tools, here referred to as annotation systems, are required to generate and enrich metadata for a video. We classify these annotation systems according to their *interaction level*, *accuracy level*, and *semantic level*, as shown in Figure 2.2. The interaction and accuracy levels determine how the annotations are generated while the semantic level is used to specify what we can annotate with a system. Moreover, annotation tools can be distinguished by the *purpos*e of the annotations, their *functional specifications*, and the used *modalities*.

The interaction level indicates the amount of human efforts in the annotation process. Automatic annotation systems work without interaction, but their development costs are usually high and the results have a lower quality than interactive annotations. However, advanced audio and video processing technologies are especially exploited in automated systems and they are often highly customized in order to achieve acceptable results for specific video tasks, domains, and datasets.

As shown in Figure 2.1 and Figure 2.2, different types of metadata can be generated on different accuracy levels. The majority of annotations are given *globally* for the entire video while more accurate annotations are given together with the *temporal* information that indicates at which time point (video frame, shot, or scene) something happened. Highly sophisticated annotation systems additionally include *spatio-temporal* information to identify at which position (bounding-box) certain objects are placed. Videos are usually annotated to enable video *retrieval*, to enhance the *visibility* and popularity of shared videos, or to perform video *analytics*. Moreover, annotation systems can operate on *visual*, *audio*, and *text* modalities. The functional

	Image	***Audio***	***Video***	***Metadata***
Content	JPEG, GIF, TIFF, BMP	WAV, MPEG-1, MP3, ADPCM	MPEG-4, AVI, QuickTime,	SMIL, MHEG-5
Metadata	EXIF, IPTC PM, DICOM	MusicXML, ID3	MPEG-7, LSCOM	DublinCore, MXF, MPEG-7, NewsML
Retrieval	JPSearch			MPEG Media AF
	ANSI/NISO Z39.50, Common Query Language			

Table 2.1: Content, metadata, and retrieval standards.

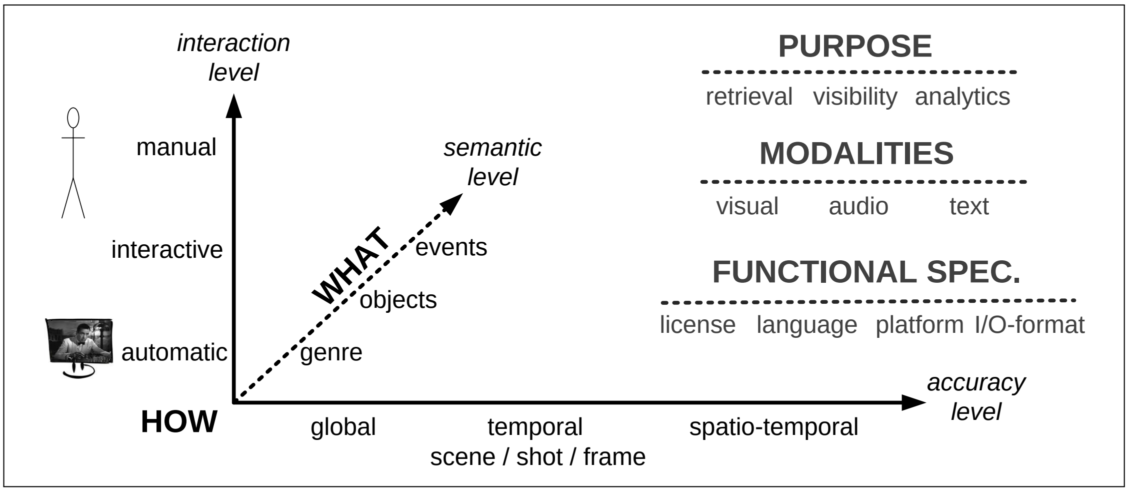


Figure 2.2: Distinguishing features of annotation tools.

specifications of an annotation system consider the available *licenses*, the computer *language* in which it was developed, the operating *platforms* it can be used with, the supported input video *formats*, and the output format of the resulting metadata. [208] gives a comparison of popular annotation systems according to these functional specifications.

Automatic Annotations

Automatic annotation techniques are mainly focused on the generation of additional, richer metadata that is suitable for text-based video retrieval. Such techniques stem from several research directions and they operate on different modalities [285]. For the *visual modality* there are shot detection tools, scene classification, object recognition, people and face recognition, setting detection, optical character recognition, concept, event, and attribute detection as well as near-duplicate detection tools. *Audio content* is analyzed with techniques for automatic speech recognition, transcript generation, speaker identification, sound classification, music recognition, as well as rhythm and timbre detection tools. Moreover, sources like web-page related text, program guides, and the date, time, and geo-codes of recordings and broadcasts are commonly used for metadata extraction. Some of the mentioned techniques are already integrated into commercial products although most of them are still in an experimental state.

The simplest visual annotation systems are used to classify entire videos into a few predefined *genres*, such as sport, news, music, and action [36]. More sophisticated approaches annotate videos with up to a few hundred fuzzy *concepts* [331]. In this process, objects and object relations are given as concepts together with events and sometimes even with attributes like the size and speed of objects [160]. In order to achieve temporal and spatio-temporal annotations, videos are first segmented with *shot boundary detectors* that already achieve high detection rates for hard cuts and gradual transitions [168]. After this, scene detection or classification techniques can be applied in order to annotate specific scenes, as it is done for action scenes in [48, 174, 187]. *Keyframe extraction* is another basic technology that is widely used by automatic annotation systems and that is integrated into many products [54]. The keyframes are thereby either randomly sampled or, in more advanced approaches, extracted according to the shot structure and change ratio of a video. On the one hand, most annotation systems provide user interfaces with a compact keyframe representation to visualize videos and to summarize them. On the other hand, many automated annotation systems only analyze the extracted keyframes instead of every video frame for efficiency reasons. Similar to video summarization by keyframes, image collections are often summarized by those images that represent the most interesting visual content based on visual clustering and keyframe selection [244, 255, 282].

Many scene-based and shot-based annotations originate from video annotation methods that operate on *objects*, as described in the recent study of [337]. [283] proposed one of the first automatic attempts to annotate identical objects for retrieval. Moreover, [84] automatically annotates the shown actors in professional videos whereas moving objects are, for instance, captured in [78] by static cameras. Saliency detection methods [135, 188, 291] are used to identify which regions of an image are likely to contain a foreground object, and frequently occurring regions are automatically detected in [145] and [254]. Although a lot of work has been done to improve object recognition in the last decades, resulting techniques are mainly integrated into annotation systems that stem from research prototypes. Only face detection and person detection [59, 327] are frequently used in products like video cameras. Furthermore, setting detection and optical character recognition techniques are supposed to provide a high potential for commercial applicability. These tools try to identify the location (city, building, street) where a video scene is recorded and to extract text that is displayed either as overlays (e.g. subtitles) or on shown objects (e.g. from the license plate of a car).

Beside object recognition, *event* and *attribute* detection [90] are popular research directions for automated video annotation. Events are the semantic concepts that humans perceive when they observe a video sequence, they happen at a given time and a given location. [167] compares a large corpus of existing work in the area of automated event detection that is extended by [20] for the annotation of edited and unedited video. The most common events that are annotated are human actions, and the survey of [252] indicates that these systems already achieve respectable results that might reach marketability in the near future. Thus, a lot of research focuses on human behavior and crowd analysis [142, 155]. Moreover, violence is an important event type for the news domain and some automatic annotation approaches have been proposed [66, 184].

Another trend in computer vision is the use of existing annotations for the generation of richer metadata. Similar images or videos are thereby obtained by content-based retrieval (compare Section 2.1.3) to collect additional annotations for the investigated content. *Search-based annotation* approaches [21, 152, 334] are further used to collect a sufficiently large training set for machine learning environments, for instance in the interactive annotation system TubeTagger [316]. Moreover, [21] exploits the social knowledge that is embedded in Wikipedia [340], Flickr [94], and YouTube to generate annotations without user interaction. [280] proposes an approach to employ the redundancy of online video portals in order to improve the annotation of videos. In this approach, near duplicate video detection is performed to detect identical and overlapping videos that have been uploaded to exchange their annotations. From a copyright detection perspective, content-based copy detection approaches are interesting for commercial products and online video portals [168]. Thereby, identical videos that might have undergone some digital transformations are detected without the use of watermarking and they are prohibited from a portal if they violate some copyrights.

Manual and Interactive Annotations

Manual and interactive annotation tools enable users to associate different kinds of metadata to a video document. The interaction level of a tool depends on the functionality it supports and on the provided interfaces. Very simple tools only support text input that is globally connected to a video as title, description, or label. Examples of such manual annotation tools are the video upload interfaces of web video portals. Advanced systems like Viper [208], the Shot-Tagger [332], ELAN [345], and ANVIL [153] allow temporal and spatio-temporal connections of the annotations to a video and aid their generation with user-friendly interfaces and computer vision technologies. The user study of [50] indicates that video retrieval with such sophisticated annotations is more convenient for users than the retrieval with global annotations.

A major factor that distinguishes the interaction level of an annotation tool is its use of *computer vision* techniques and *semantic web technologies*. Viper [208] supports domain specific extensions and computer vision to perform activities like text detection and face recognition. Similarly, [111] and [110] perform the detection and tracking of moving objects as offline step to ease their annotation. [61] presents a recent survey about video annotation tools that focus on the use of semantic web technologies for the annotation process. Domain ontologies are thereby proposed as a major tool to generate semantic annotations because they exploit the use of a common vocabulary with a fixed interpretation. For instance, the Arneb system [7] facilitates such annotations with a video ontology and stores the output metadata in MPEG-7 and OWL.

The interaction processes that are required to generate specific metadata are another difference of existing annotation systems. [328] presents a manual system where users can make comments to spatio-temporal regions of online videos. Objects, object relations, and object related events are then automatically extracted from these user comments to enable object-based and event-based scene retrieval. Similar object annotation systems are given in [335] and [216]. An interactive video segmentation was presented in [336] to extract objects-of-interest from the background for annotation. In addition to textual input, graphical input is supported in some tools. [110] presents an object annotation system that can be used to annotate football players and their moves in a play using graphical annotations. [111] allows five different types of graphical input including graffiti, scribbles, word balloons, and video hyperlinks to annotate moving objects. The Arneb system [7] can be used to insert simple notes, speech bubbles, and sub-titles directly into the video content. SeViAnno [258] further allows the annotation of locations and geological coordinates by a single click on a Google maps frame.

Different annotation systems have been developed for different purposes and [61] concludes that the choice of a system mainly depends on the intended context of usage. The systems LableMe Video [361], Viper [208], and Arneb [7] are ground-truth authoring tools that try to facilitate the development and evaluation of automated analysis methods. Thereby, information about objects (class, shape, motion) and their activities is generated. Domain specific systems, such as SeViAnno [258] for the cultural heritage domain, VideoAnt [131] for the peer reviewing process of educational videos, and the eSport system of [363], are tailor-made for their field of application.

Personal video collections are usually annotated by *single-user* systems while videos that are publicly available or shared between several users can be *collaboratively* annotated. However, in most video portals only the user that uploads a video can set the title, a description, and some tags. Other users are only allowed to comment the video and to judge it. In contrast, every user can participate in the annotation process of the research prototype SeViAnno [258]. The Synvie system [354] goes even further and automatically associates user comments that stem from conversations about a video within messaging systems directly to the scenes of this video. Moreover, [129] proposes another collaborative video annotation tool that especially focuses on a software architecture that allows change tracking in order to keep all collaborative annotators up to data. In contrast to collaborative annotations, users can add personal notes to music items in the Last.fm portal [166] in a way that they are visible to no other user.

The required annotation time that a user has to spend with a system is another factor that should be considered in the system selection process. Real-time annotation systems, such as the eSport system [363], allow the generation of metadata when the user watches a video. Accurate annotations of a video that are generated with usual systems (including play, pause, backward and forward video controls) can take much longer, especially when the spatio-temporal extent of objects and events are annotated on frame-level. Another aspect of annotation systems is the particular time at which the metadata is generated. While most systems are meant to annotate videos that are ready to be viewed, it is the main goal of DiVA [119] to gather annotations during the creation of video sequences and animations with video editing and post-effect tools.

User Intentions

The annotation of video documents is motivated by the similar goals as the metadata generation for traditional web documents. Generally, videos are not annotated to enhance the video description or to enhance the searchability of video archives but to improve the video popularity [105]. Video portal users mainly want to achieve high visibility of their content and try to promote their videos as good as possible. Thus, very general titles, descriptions, and user tags are often used that match commonly used search terms, such as the title ’funny video’. Moreover, [47] shows that videos generally receive more views when they are annotated with more information, using longer titles, descriptions, and more tags. [10] further states that users have a higher motivation to annotate their content when they get the right tools for annotation. However, temporal and spatio-temporal annotations are time-consuming and difficult to generate even when approved annotation tools are applied. Consequently, these kinds of annotation are only generated for good reasons: In order to achieve scene-level retrieval within small video archives, for video analytics especially in the sport domain, or to generate ground-truth and training data for automated video analysis [361]. Annotation games, such as OntoTube and Yahoo’s VideoTagGame [299], exploit another motivation to annotate videos. In these games, users compete against each other. They collect points for fitting annotations, and try to get on top of the highscore tables. [127] further shows that users are more willing to annotate videos if they are interested in the video content of these games.

Some work has also been done to understand the annotation process of users and the resulting annotations. Collaborative annotation systems are described in [201] together with a taxonomy to compare different architectures, as it was done for Flickr and Delicious [67]. [109] pointed out

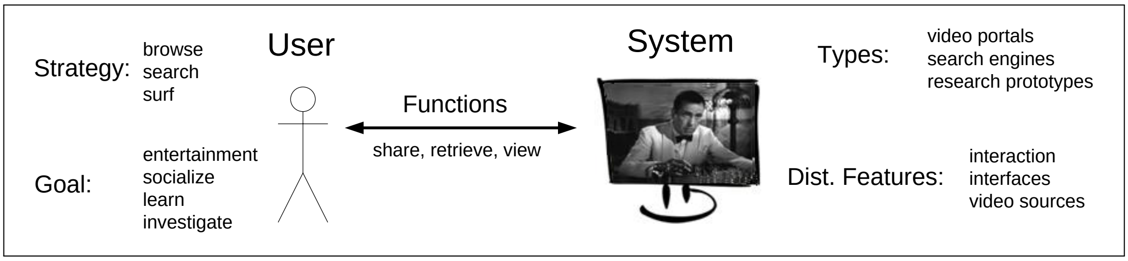


Figure 2.3: Video retrieval systems and their users.

that there is a large variation in the number of annotations per user of collaborative annotation systems. Heavy-users generate hundreds of annotations for their own content and the content of other users, while there are many users that only annotate their own content with very sparse descriptions. Moreover, [75] investigates why users give specific tags to an image. In this study the regions-of-interest where marked and rationales could be given to them. The work of [77] investigates user-generated tags over time and they mention that some annotations are temporally correlated to specific events, such as the London marathon. Associated annotations are mainly generated shortly after the actual event. Similarly, [123] studies the distribution of user tags in collaborative annotation systems over time and analyzes the meaning of high-frequency tags using the relation to other tags. [109] further classified the used tags according to their functionality and [281] associates Flickr-tags to WordNet [348] categories in order to identify what users are annotating.

2.1.3 Video Retrieval

Although video sharing and retrieval are very popular nowadays, the used technologies are still in an infant stage compared to the technologies that are used in search engines for the retrieval of text documents and websites [366]. In contrast to text documents, the semantics of most videos are not explicitly known and videos exhibit a larger content richness. As a consequence, video representation is difficult and users need more time to validate if a returned video appropriately answers their search queries, to navigate through videos, and to retrieve specific parts of a video. These facts make today's video search to a highly fragmented and sometimes frustrating experience [285] and they make it difficult to evaluate how successful video retrieval systems really are in terms of effectiveness, efficiency, and flexibility [104].

Figure 2.3 gives an overview of video retrieval systems and their users. The right side shows the different *types* of retrieval systems that operate on the video metadata and, to some extent, directly on the audio-visual content. These systems differ by the supported interaction types, their user interfaces, and the used video sources. The left side of the figure shows the used *strategies* and *goals* of typical users that want to share, retrieve, and view specific videos.

System Types

As shown in Figure 2.3, we distinguish between three different types of retrieval systems: *online video portals*, *search engines*, and *research prototypes*. Generally, all systems aim at the maximization of search queries and video views in combination with a minimization of undesired media streaming initiations. Video portals, such as YouTube [359], Hulu [137], and Daum UCC [63], further try to provide as many videos as possible, they store the video content on their servers and enable users to upload, annotate, and judge these videos. Moreover, they provide social services to connect users to each other. On the one hand, video portals support video retrieval by Google-like *text searches* and the recommendation of related videos. On the other hand, it is possible to retrieve the videos of most portals through *external links* that might follow postings in social networks, e-mail recommendations, or that stem from search engines. According to [64], *internal links* account for about 60% of all views in YouTube. The presented recommendations are personalized for each user and they reflect her recent activities on the site. Related video links are shown for videos that stem from the same user of a selected video, that are similarly tagged, or that share other similarities like internal user recommendations or comments. [47] states that up to 30% of all YouTube views stem from external links and that 47% of all videos are linked in external websites. Such external links provide a way of advertising the videos, and thus they are especially important for recently uploaded videos in the first days on a video portal [159]. Complex server architectures and service models are used [50] to deal with the vast amount of data requests. Portals that focus on user-generated videos, like YouTube and YouKo [358], mainly rely on advertisement as their source of revenue while paid accounts are frequently used from domain specific portals, such as Netflix [227], where only professional content is shown.

Video search engines, such as Videosurf [326], MetaCrawler [206], and PolyMeta [250], return videos from several destinations on the web but do not store or provide these videos on their own, similar to search engines for usual web documents. Like video portals, they are mainly funded by advertisements that are shown beside the retrieval results. Most traditional web search engines including Bing [28] and Baidu [17] provide video retrieval facilities from different video sources while Google's video search mainly returns videos that are stored on its own video portal, YouTube. Another kind of search engines is specialized on professional video content, such as IMDB [140], TheMovieDB [300], and AllMovie [9]. These engines offer information about movies and TV series and provide links to buy the actual content. Furthermore, some video search engines return videos that are shared in peer-to-peer networks. In these searches, videos are mainly identified by their title which often leads to an unsatisfying retrieval experience.

The third type of video retrieval systems is given by research prototypes. These systems have a different scope than video portals and search engines. In the first place, they try to propose novel retrieval technologies that might be used in the near future and demonstrate them on small datasets instead of enabling video retrieval on existing databases. A lot of these systems have been presented in the 90s, like the QBIC system [93], VideoQ, and VARS [362]. Recent systems are the MIRACLE video search engine [107] from AT&T Labs and MARVEL [226] from IBM Research that leverages multiple content modalities in combination with all kinds of metadata. Moreover, a few systems have been proposed that are specifically designed for large-scale video collections with a focus on scalability [216, 237, 352]. The systems of [194] and [27] investigate different feedback and query expansion strategies.

Interaction

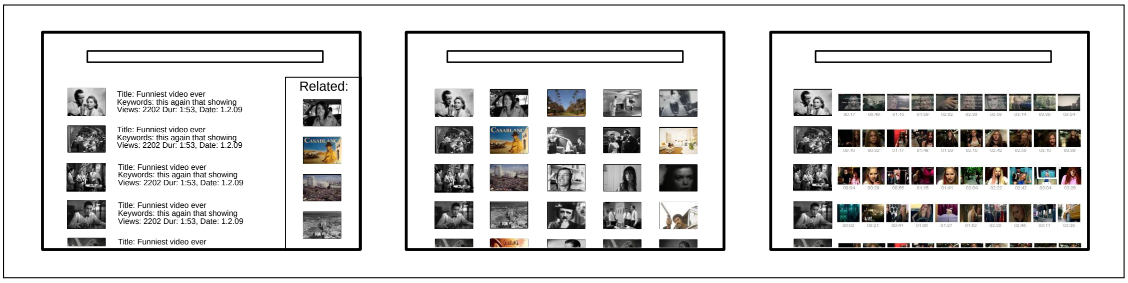
There are different ways to access videos on the web. From a traditional web search perspective, Google-like *text queries* seem to be the most natural approach to retrieve videos that fit to some keywords while *content-based queries* might be used to retrieve videos that are similar to a given example. More than a decade ago, studies about the formulation of text-based video queries have been made in [262] and [30]. They distinguished between queries about the videos as a whole; queries about the topic content; queries about sensory content like the appearance or location of objects; and queries about the data and metadata. Beside keyword-based text queries about abstract video concepts, well-defined query languages have been proposed to express complex statements [353]. For instance, [257] models spatio-temporal information between different objects for video scene retrieval while a couple of research prototypes make use of Allen's temporal relations [8]. Moreover, [298] developed a system that enables the use of natural language queries for the retrieval of specific clips in surveillance videos using queries like 'find scenes where people are moving across an island or along the right side of the island'.

Content-based retrieval techniques are less popular for video than for image and audio retrieval. For instance, the mobile phone apps Shazar [276] and Goggles [108] facilitate their users to identify which song they are listening to or which painting they are looking at. These services are based on near duplicate detection. Similar-content retrieval is mainly popular for images ever since own images can be used as queries for Google's images search. The predominant inconvenience of content-based query systems is the query generation itself, as it is often difficult to obtain content examples that fit to the specific retrieval goals a user might have in mind. [366] further points out that text-based queries are more accurate and convenient because it is not clear which parts of a content-based query are interesting for the user and how they should be interpreted. According to [79], there are three different interpretation levels for content-based retrieval queries. At the first one, a user tries to retrieve content like the one she starts with. The second level is the retrieval of individual objects and object classes that are shown in an image or video. Finally comes the retrieval by abstract attributes that often depicts the purpose of objects or scenes, including queries about specific events and about the emotional state.

Differences that concern the interaction with retrieval systems further originate from the way answers are generated and refined. Most video retrieval systems provide *refinement* or expert search opportunities to set some filters, for instance, to restrict the results to one of a few predefined genres, the video duration, the content creation and upload date. In addition, search engines sometimes allow the selection of video sources and content types. Typical search engines return the same results for the same query if they are submitted by different users whereas the personalized approach of [366] automatically sets a category filter for each query, based on the user's search history. Moreover, relevance feedback algorithms are often used in research prototypes to maximize the distance between relevant videos and non-relevant but similar videos in the result list [104].

Interfaces

Today's video retrieval systems try to answer user queries with similar interfaces that are used for web documents [107]. In this process, a set of videos are returned for a query and the results



Title: Funniest video ever
Keywords: this again that showing
Views: 2202 Dur: 1:53, Date: 1.2.09

Title: Funniest video ever
Keywords: this again that showing
Views: 2202 Dur: 1:53, Date: 1.2.09

Title: Funniest video ever
Keywords: this again that showing
Views: 2202 Dur: 1:53, Date: 1.2.09

Title: Funniest video ever
Keywords: this again that showing
Views: 2202 Dur: 1:53, Date: 1.2.09

Title: Funniest video ever

Figure 2.4: Retrieval system interfaces (left to right): classic, compact keyframes, time line.

are presented with one or multiple keyframes and some metadata, such as its title, duration, number of views, and a general description. Figure 2.4 presents three abstract interfaces that are commonly used to present the retrieved videos. The *classical view* is adapted from web document retrieval systems. On top of the page a search query can be entered in a text box. The results are presented in a list, one below the other, using a keyframe and some metadata. Usually, the same keyframes (e.g. the first frame of a video) are used to present a video independent of the submitted query [315]. However, the same interface can be used to play the videos within the keyframe widgets, as it is done in MetaCrawler. The free space on both sides of this classical view are often filled with advertisements and recommended videos. The *compact keyframe view* presents many videos on one screen which is, for instance, used in PolyMeta. Metadata about a video are shown there as tool tip when the mouse cursor is situated above a keyframe. The third interface is the *time-line view* that is exploited in VideoSurf. Multiple keyframes are used in this view to present each video in one line and the timepoint of each keyframe is displayed below it.

Most video retrieval systems offer *browsing* facilities based on the current search results and the stored preferences of a user. Related videos are thereby offered, as shown in the classical view of Figure 2.4. The research work of [289] explores new video browsing techniques in a 3D keyframe view that allows the visualization of more videos on one screen than the compact keyframe view. Furthermore, they propose a zooming functionality to enlarge the thumbnail that is shown under the mouse cursor and show different frames of this video when the mouse wheel is scrolled up or down. VideoSurf [326] provides another innovative browsing facility in order to navigate between different videos that contain the same person. In this interface, images of all annotated persons are shown for the current video, and new videos are loaded after a user clicks on one of these persons.

User Intentions

Users that want to retrieve videos on the web follow the same intentions as users of traditional broadcast television. On the one hand, they browse through videos with the simple, unarticulated intention to be entertained by content that they find interesting, similar to TV users that are zapping through the channels. On the other hand, they are interested in specific information and content. According to [62], users can be classified as *browsers* with no clear end goal that perform a series of unrelated searches within a session and jump across multiple topics; *surfers* that have a moderate clarity of an end goal. Their actions may be somewhat exploratory in the beginning but increase the clarity in subsequent searches; and *searchers* that are very clear about what they are searching for in the system and only need short sessions to retrieve an end result. [200] further distinguishes between users that *search* for specific items (e.g. a movie they have heard about), users that want to *learn* something in order to develop new knowledge, as well as users that want to *investigate* something. The two later classes perform video retrieval with multiple search iterations, but the investigative search iterations are distributed over long time periods. Similar to video retrieval, there are different reasons why users want to share video content. Some users produce content by themselves and like to share this content with other people. Another group of users try to promote videos that they like.

The work of [124] investigates how people search for video documents and they concluded that users typically prefer short queries, only look at a small number of results, and seldom modify the query. In addition, [57] performs a video retrieval study in the wild, where users are asked to search for videos as they do it in their everyday use. In this study, video retrieval on YouTube accounts for two-thirds of all sessions in front of search engines, other video portals, TV and commercial movies sites, domain specific sites, peer-to-peer services, and social networking sites. For about 80% of all video searches only one web site was consulted.

According to [262], users adapt to the provided technologies, and thus there exists a relationship between the most used query types (about videos as a whole that can be answered with bibliographic metadata) and the availability of results. In other words, this means that people tend to ask queries where they expect reasonable answers. [124] pointed out that most web users regularly use video platforms but they do not necessarily contribute to these portals. Moreover, it has been shown that there is an extremely weak social connectivity between users of a video portal [49, 180] and that users are generally not biased to rate popular videos more often than unpopular ones [47].

2.2 Object Recognition

Computer vision systems recently achieved fairly good results for the recognition of individual objects [191,211] and object classes, for example, in the Pascal VOC challenge [85]. One factor that accounts for this success is the large amount of research that focused on object recognition during the last years [58, 80]. The published amounts of object recognition papers that are submitted to the leading conferences in the areas of computer vision and pattern recognition approve this trend. In this section, we first discuss why object recognition is a difficult task at all before different application areas and the main principles of automated recognition systems are given.

2.2.1 Objects in Videos

Object recognition can be defined as the ability to discriminate known objects (*identification*) or a set of objects (*classification*) from other objects, materials, and textures at different levels of specificity [69]. A dog can be recognized as ’my dog’ on the individual level or as ’golden retriever’, ’dog’, ’mammal’, and ’animal’ on the categorical level. Although object recognition seems to be very easy for humans, it is difficult for automated systems because the same object

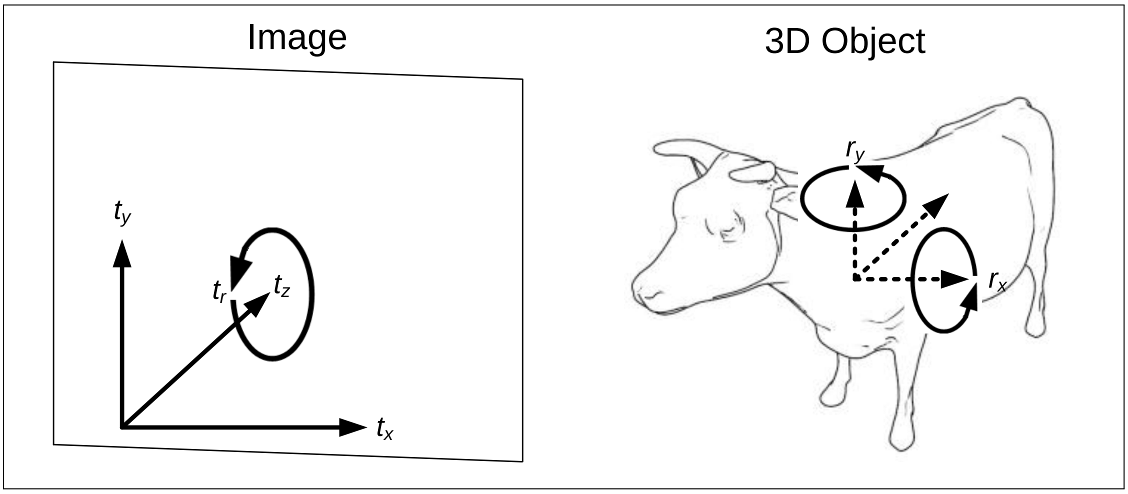


Figure 2.5: Translational and rotational degrees of freedom that affect the appearance of an object in the image plane.

can appear very dissimilar in different images and videos [217] according to following five categories of possible change [270]:

Similarity transform: As shown in Figure 2.5, an object has three translational degrees of freedom (t_x, t_y, and t_z) and one rotational degree of freedom t_r in the image plane. The position of an object changes if it is translated in x or y direction. A translation in z direction changes the size of an object, and thus the object becomes bigger or smaller in the image and more or less details become visible. The translation around the t_r axis changes the 2D pose of an object although exactly the same object parts are shown. However, after such rotations an object might be standing on its head.

3D transform: An object has two additional rotational degrees of freedom (r_x and r_y), as shown on the right side of Figure 2.5. A transform around these axes heavily changes of the appearance of complex 3D objects because they look different when they are viewed from the front, the back, the bottom, or the top. On the other hand, objects are less distracted by 3D transforms (e.g. caused by small perspective changes) if they only consist of a planar surface, like books and cereal boxes.

Scene changes: An object can appear differently because of other objects that are contained in the same scene. On the one hand, objects can be partially occluded by other objects and they can partially run out of the image. On the other hand, objects are easier to recognize if the background contains different colors and textures than the object itself.

Light conditions: Objects appear differently when changes of the intensity, color, or direction of the light source that illuminates them happen. Illumination effects, such as shadows and reflections, have a further impact on the object's appearance.

Imaging conditions: Different types of signal disturbance like noise, quantization errors, and blur can decrease the quality of images and videos in a way that influences the object appearance.

Mold and other symptoms of old age as well as digital image and video artifacts provide another source for appearance changes.

In addition to these points, individual objects can change their appearance over time. For instance, TV screens and chameleons are able to change their colors, scissors can change their shape by a rigid deformation while faces are deformed in a non-rigid way when their expression changes. As cited in [84], persons further exhibit variations in their visual appearance due to changes in hair style, clothes, and aging. The body shape of persons changes if they lose or put on weight, train their body, undergo plastic surgery, or if visual effects are applied to images of them with touch-up tools like Photoshop [248].

The recognition of object classes is additionally difficult because the objects of a class can differ in multiple ways and their common features might be hard to recognize for automated systems. Some object classes are, for example, mainly defined by their functionality (chairs for sitting, and coffee cups for drinking) and objects of these classes can differ in form, color, and size [115]. For the definition of this kind of objects, the function (what someone can do with them) is the most important feature. In the context of computer vision systems, [97] distinguishes between *things*, objects with a specific size and shape, and *stuff*, materials that are defined by a homogeneous or repetitive pattern without explicit spatial extents and shapes, in order to use different recognition strategies for them.

Beside the actual appearance of objects in images and videos, their surrounding provides an important input cue to ease recognition. Research from cognitive psychology shows that the presence of a particular object constrains the identity and location of nearby objects, and that visual systems can make use of this context information for object recognition [232]. The probability of an object to be contained in some kind of scenes (*semantic context*), the probability that some objects are situated in specific geometric constellations next to each other (*spatial context*), and the likelihood of an object to appear at certain sizes in a scene (*scale context*) are thereby discriminated in [102]. [70] adds further context types, such as photometric context, illumination context, weather context, and geographical context to gather information about cameras, lenses, the direction of light sources, the temperature, season and wind speed, as well as GPS location, terrain type, and population density. According to the way in which context is extracted by visual perception systems, global and local context can be distinguished [102]. In the global variant, the gist of an entire image or scene is observed at once whereas local context works on an object, region, or pixel level.

2.2.2 Application Areas

Both, humans and computer vision systems, perform object recognition from two opposite directions. On the one hand, *top-down* recognition is done to identify specific objects and object classes that are known in advance within an image or video, as described in the next section. On the other hand, *bottom-up* recognition is done to select regions that might contain an object based on the saliency of the image without prior knowledge about the scene. Such saliency-based visual attention models are suitable to detect unknown objects [265]. The visual attention is thereby obtained by accentuated colors or shapes of an object that stand out from the background. When it comes to video, motion is an excellent saliency feature. Motion is especially exploited in surveillance applications, where static cameras are used to monitor indoor and outdoor locations in order to detect objects (e.g. persons and cars) that pass the scene. As the background of these locations only undergo minor changes due to the light at day and night times, simple techniques are often sufficient to detect moving objects although sophisticated approaches also exist, see [44]. However, the same task is difficult if moving cameras and arbitrary scenes with more heterogeneous objects-of-interest are given. [173] tries to solve this task by an automated foreground object segmentation that achieves promising results.

In addition to object recognition, several approaches have been proposed to extract visual attributes like the color, patterns, and the shape of objects. For instance, [90] tries to recognize that various objects can be *yellow*, buildings can be *tall*, clothes *plaid* or *striped*, and wheels *round*. Similarly, [86] proposes an computer vision system that tries to describe the visual content of an image by its attributes instead of identifying the contained object classes. In this approach, unknown objects might be correctly described (e.g. by the number of legs, the nature of their coats, and their sizes) although the system have never seen an image of such an object or animal before. Known objects are further classified by this system according to their attributes. Moreover, [115] proposes an affordance detector to classify the function of investigated objects (e.g. that chairs, benches, and sofas are used for *sitting*) while [190] determines human actions (walking, jumping, or swinging a golf racket) from attributes. The work of [156] goes one step further and proposes a system that automatically explains the content of images in simple sentences (e.g. 'This is a picture of two dogs. The first dog is near the second furry dog.') using recognized object classes, their spatial relation, and object attributes.

Object recognition systems can be distinguished according to their recognition level. The simplest systems only determine the existence or absence of an object in given images or videos. More sophisticated systems compute the spatio-temporal position of the objects and try to identify the current pose of these objects. Object segmentation approaches try to compute very accurate object boundaries in a way that objects can be cropped out of an image or video to reuse them in other content. Object tracking, on the other hand, is often done for high-level applications where the actual position, size, and pose of an object is required in every video frame [357]. Depending on the accuracy requirements of an application, rough bounding boxes around an object, point clouds, or accurate object silhouettes are used in these tracking applications.

Section 2.1 describes the use of object recognition for image and video retrieval systems. In addition, there are many other applications where object recognition is used as input cue. First, industrial applications use object recognition for position measurement, inspection, sorting, counting, and related tasks [307]. These applications are common in automated production processes of manufacturers, for example, to inspect the surface of industrial parts to detect defects. Second, applications in the area of robot navigation [31] and surveillance [134] often rely on object recognition to perform tasks like access control, person identification, congestion analysis, and anomaly detection. [364] reviews popular face detection approaches that are commonly used in these tasks, such as the popular Viola-Jones face detection [327]. Face recognition techniques are already used in real-world applications and products like digital photo cameras and photo sharing services, see Picasa [249]. Gestures, emotions, and physical attributes of persons are captured in [178] to predict the social behavior of communicating persons whereas social behavior patterns are automatically detected in [35] to identify events like fighting, vandalism, and overcrowding. The recognition of human actions [3] is also important for assisted-living applications to support elderly and blind people in their everyday tasks.

Object recognition is further used for sport analysis, as described in [330]. Beside tactic analysis and automated highlight generation, there is an emerging request for systems that can assist the referees at run-time in order to decide the position of sport equipment (goal or no goal, in or out) and players (off-side situation in soccer). [193] performs an automatic identification and tracking of basketball players in usual TV broadcasts with results that are sufficient to enable sport analysis systems, for instance, to investigate the strength and weaknesses of opposing teams and players. Moreover, the recognition of moving objects can be used to improve video coding and compression, especially for static cameras where the background does not change over long time intervals [44]. Augmented reality applications are another application area where recognized objects are used as foundation to insert artificial overlays in form of simple text and complex 3D animations [103, 112]. As concluding remark we would like to mention that scene understanding is sometimes referred to as the ultimate goal of computer vision research [122, 178] and that object recognition is an important technique to solve this task.

2.2.3 Recognition Techniques

As noted in [222], research on object recognition has been intensively practiced for five decades. In the first decades, the focus was set on the development of analytic representations that are suitable to model the appearance of broad object categories from different viewpoints and under any illumination conditions. In these *geometric models* objects are presented by structural information and 3D properties (lines, edges, and ellipses) using computer-aided design models. Unfortunately, such CAD models are only available for a few manufactured objects and they cannot capture complex objects like trees or planar objects that are only distinguished by their color and texture, such as paintings. Therefore, an alternative approach that directly uses the *appearance* of objects in images emerged in the late 90s with the groundbreaking work of [271]. These recognition approaches uses the pixels that are actually shown in the 2D images to extract the color, texture, and shape of an object. The simplest approaches generate such object models globally from entire images whereas more sophisticated approaches are based on a collection of viewpoint-specific local features [296].

Color, texture, shape, and motion are the visual attributes that can be used to extract features for object recognition. In this context, good visual features should have the same values when they are computed from the same objects and distinct values for different objects. In addition to this *distinctiveness*, following factors determine the quality of a feature. *Repeatability* is important to extract the same features in different images of the same object while the *quantity* assures that a sufficiently large amount of features is extracted from every object in a way that object recognition succeeds even in the case of object occlusions. *Invariance* and *robustness* against certain image and object transformations (see Section 2.2.1) are necessary to reduce the required number of training images and the learned object poses, and light conditions. Therefore, small perspective changes and variations of the illumination do not affect the distinctiveness of an invariant feature.

The methods which are used for object recognition vary from one recognition system to the other. It is well established that feature types mainly differ by the trade-off that they achieve between their discriminative power and invariance. Different recognition tasks require different trade-offs, and thus no single visual feature is optimal in all situations [321]. In addition to visual features, [146] has shown that all components of a recognition system can have strong influences on the achieved results. Popular approaches include interest point and region detectors [212], segmentation-based approaches [45], dense sampled regions [268], and it was also shown that global approaches can be very efficient for some tasks [231]. Texture-based features [211] are mostly used to describe local regions while color-based and shape-based features have been of minor interest in the last years. However, the success of combined features that incorporate color and texture [268] indicate the importance of all feature types. After extraction, visual features are used to form object models like the popular bag-of-features [170] and sparse representations [197]. These models are then matched against the features of an image or video using different matching strategies or machine learning approaches [132]. At last, a verification step is often used to validate the geometric constellation of different features and objects.

As described above, most computer vision systems perform object recognition with the steps feature extraction, matching, and verification in one single iteration. Humans, on the other hand, constantly switch between bottom-up and top-down recognition to identify objects in a scene or image [296]. For this reason, an iterative object recognition technique is presented in [89] where expansion and contraction steps are executed one after the other, and in [239] where a hierarchical object part-model is used. In the latter approach, low-resolution parts are first matched against an image to propagate possible geometric constellations for the high-resolution parts. This reduces the required efforts for matching significantly compared to brute-force approaches as only a few image parts have to be matched against every high-resolution part. Moreover, [172] proposes a method in which the most discriminative objects are recognized first in an image. In each iteration of this process, the system tries to recognize the next easiest object category. The use of context models can similarly decrease the number of object categories, positions, and scales that have to be considered in the recognition process because objects never appear in isolation in the real-world but in the context of entire scenes [232]. [102] presents a recent study about the use of such contextual information for object recognition. They found that although current context models are able to improve recognition approaches, they lack of complexity and scalability issues when the number of object classes increases. External sources of context are further required to provide information about a sufficiently large number of objects and scenes in order to aid real-world applications.

In conclusion, it can be noted that the success of recognition systems is affected by different criteria including the used object models, the training data, the expected object variations, the quality of the content data, and the used matching strategy [307]. The study of [238] further invested why computer vision performs much worse than the human visual system with emphasis on the three factors learning algorithms, training data size, and visual features. In these experiments no evidence was found that the human learning algorithm performs in any sense better than modern machine learning algorithms or that humans leverage a larger corpus of training data to increase their recognition performance. Thus, the authors hypothesized that the used visual features are the critical factor that gives humans an advantage over machines.

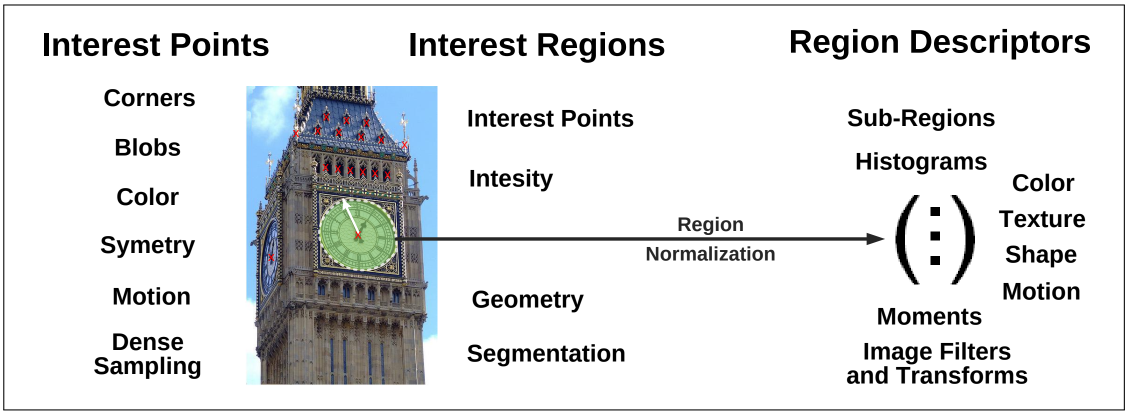


Figure 2.6: Generation of interest points, interest regions, and region descriptors shown together with their distinguishing features.

2.3 Visual Features

Local features [211] are an integral part of the best practice in object recognition. These features are either computed from dense sampled regions [268] or from automatically detected interest regions that are often situated around interest points [212]. Despite the simplicity of dense sampling, such regions are commonly used for object class recognition whereas interest region-based approaches are predominantly used for individual object recognition. The resulting regions of both selection techniques can vary in shape and size, and usually a set of many overlapping regions are selected from each image. Descriptors are then extracted from these local regions to capture color, texture, shape, or motion information of an object. In the following, we describe the state-of-the-art of interest points, interest regions, and region descriptors.

2.3.1 Interest Points

Schmid and Mohr [271] introduced the term *interest points* to the field of object recognition in the late 90s. Different *detectors* are used to generate such points around saliency properties of an image or video frame. The amount of interest points in an image depends on the used interest point detector and the visual richness of the image [269], and a good distribution of the interest points over all shown objects is important to achieve appropriate recognition results. Moreover, these points have to be *stable* and *repeatable*, which means that interest points should be detected at the same parts of an object if it is shown in different images, see Section 2.2.3. Generally, interest point detectors iterate over an image to test if the neighborhood of an investigated pixel contains the wanted saliency property or not. Thus, interest points are usually located at the position of image pixels although *sub-pixel* approaches exist [38, 128] that try to capture the salience property more accurately. This is especially important when objects are shown at low resolutions. In the history of interest points, most detectors have been manually designed to capture specific saliency properties with appropriate quality and quantity for given applications. In contrast, [308] presented a genetic programming approach to select a couple of point detectors automatically in an iterative optimization-search process on synthetically transformed test images.

Corner detectors present the first class of interest point detectors that are situated at locations where the image signal changes in two or more directions. On the contrary, edge points only contain a singular direction change and they can be found at regions with a sharp brightness change, for instance with the Canny-edge detector [42]. Corner point detectors including the Harris-corner detector [126] and Lucas-Kanade tracking points [195,278] are often based on the auto-correlation function [308] whereas steerable filters are used to identifies edge foci points in an image [368]. Moreover, [260] presents a family of corner detectors that are especially designed to maximize the processing speed, as denoted by their names FAST, FAST-9, and FAST-ER. Comparative evaluations (see Section 2.5) shown that the corner detectors are very reliable in the presence of synthetic image transforms although they may not represent the most stable interesting points in real-world images [273].

The second class of interest point detectors are *blob* detectors that identify points which are darker or brighter than their surrounding. Difference of Gaussian points (DoG) [191] and the approach of Lindeberg [185] detect blobs in scale-space by a comparison of each pixel against its point neighborhood in its original scale, the next higher scale, and the next lower scale. [128] present a sub-pixel accurate blob detector that uses the Hessian matrix whereas the wavelet transform is used in [60]. A further class of interest point detectors are based on the *symmetry* of lines and point neighborhoods, as given in [274] and [192]. Both works use symmetry-based interest points exclusively for face recognition although it is assumable that they are also well suited for man-made objects that contain a high degree of symmetry, such as buildings or vehicles.

Classic interest point detectors operate on gray-level images and focus on the texture of an image. However, a few alternatives have been proposed to extend these detectors to the use of *color* in order to achieve better robustness to illumination and color changes. For instance, [114] and [290] extended the Harris-corner detector and [318] proposed a generic color boosting that can be applied to every interest point detector. In contrast, [203] presents an interest point detector that is purely based on color. This approach selects those points that consist of a certain number of different colors in their neighborhood. The extension of interest points to the *temporal* domain has also been considered in several works. The most obvious approaches use the same interest point detector in every frame of a video and select those points that are robustly detected over several consecutive frames. Point tracker, such as the popular Lucas-Kanade approach [195], follow this strategy. More sophisticated variants detect spatio-temporal interest points (STIP) using Harris-corners [161], Gabor filters [341], a 3D Hessian matrix [74], and even using dense sampling [329]. STIP points generally provide a non-constant motion over time, and thus saliency is given in the spatial and temporal domain.

2.3.2 Interest Regions

In the context of object recognition, interest regions are used to generate descriptors from images of the object-of-interest and arbitrary images or video frames where the object should be recognized. These regions should be as similar as possible when they are extracted from the same object or object class, even when the object is differently shown, see Section 2.2.3. In order to

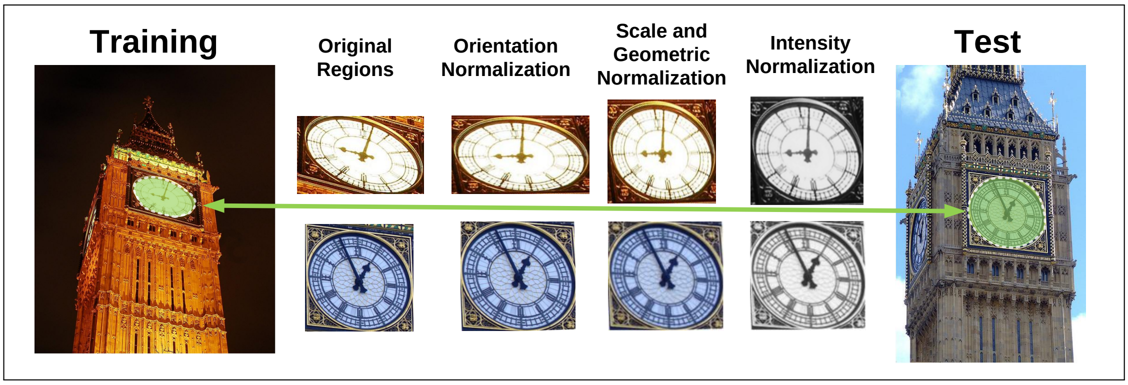


Figure 2.7: Region normalization techniques.

achieve a certain degree of invariance, either the extracted regions have to be normalized or the used region descriptors have to obtain invariant properties. During region normalization, those regions that stem from the saliency extraction step (*distinguished region*) are transformed to the regions that are finally used to build local descriptors (*measurement region*) [202], as shown in Figure 2.7. In this process, it is possible to apply scale, orientation, shape, intensity, and geometric normalizations. For instance, the pixel values of a regions are often normalized to a certain range to achieve illumination invariance whereas smoothing filters are used to extract a similar level of detail from objects that are shown at different sizes [191]. In addition to region accuracy and normalization, the original region size further affects the properties of the resulting descriptors, as small image regions seldom mix the content of different objects or from objects and the background but they contain a less descriptive power than bigger regions.

Many image regions are centered on *interest points* and one or more regions with specific sizes, shapes, and orientations are selected for each point. The original approach of Schmid and Mohr [271] uses three circular regions with static sizes around each interest point. Extensions of this approach first select the dominant scale of an interest point in order to achieve scale invariance with only one variable-sized region around each interest point. In this process, the Laplacian function [186] is used for dominant scale selection at local maximas in scale-space. An efficient approximation of the Laplacian is achieved with Difference of Gaussian (DoG) pyramids by a series of convolution, subtraction, and sub-sampling steps [191]. Amongst others, scale adapted versions of the Harris-corner detector [211] exist in this way. Additionally to the dominant scale, it is common to extract point-based regions relative to their dominant orientation in order to achieve rotation invariance. The dominant orientation is thereby defined by the image gradients around an interest point and they are obtained by subtraction of the neighboring pixels in x and y direction [143]. Beside regions that are centered on interest points, a few image regions are generated from groups of interest points. [38] forms regions from groups of two, three, or four DoG points, whereas [169] tries to select stable combinations of interest points with a maximum entropy framework. In a similar way, parallelogram regions are presented in [312] from Harris-corners and neighboring edges. Moreover, circular regions are defined by high edge-energy and entropy in [149].

Intensity-based regions try to identify connected pixels that are commonly brighter or darker than their surrounding pixels, similar to the blob-based interest points described in Section 2.3.1. These regions are considered to be invariant under affine geometric and linear photometric transformations because they automatically deform with viewpoint changes and because they are not affected by global illumination changes. The regions of [312] are constructed from intensity extrema that are detected with a non-maximum suppression algorithm at positions where the intensity suddenly increases or decreases compared to their surrounding. Maximally Stable Extremal Regions (MSER) [312] are based on a watershed algorithm to detect both, fine and coarse image structures, and they can be nested. In the same work, two region-dependent methods have been proposed to compute the dominant orientation of MSERs.

Another kind of interest regions stem from image and object *segmentation* approaches. On the one hand, image segmentation tries to identify connected regions that belong somehow together using color as main source of segmentation [207]. Object segmentation, on the other hand, tries to segment entire objects, which is difficult because most objects consist of different parts and colors that would be individually segmented in image segmentation approaches. Moreover, objects might be situated in front of similarly colored and textured backgrounds. [130] gives a good overview of object segmentation approaches including Normalized Cuts [277], MinCuts [45], and Mean-Shift [52] techniques. These approaches either work with or without knowledge about the segmented objects. In the first case, semantic segmentation [56] is used to detect and segment known objects, such as faces [327], persons [59], and cars. Approaches that are used to segment unknown objects operate with visual attention methods (e.g. [12] that starts from object contours) and objectiveness measures [5]. The segmentation of unknown objects becomes simpler when it comes to video, as moving objects might be temporally segmented from their background. The approach of [39] combines known and unknown object segmentation using a trainable object-part segmentation, edge detection, and self-similarity.

The most popular technique to achieve geometric invariance against affine transformations uses elliptical image regions and transforms them to the unit disk, as proposed in [22]. The gradients of arbitrarily shaped image regions are thereby used to fit in oriented ellipses. Harris-affine and Hessian-affine detectors [210] are proposed in this way and a combination of dense sampling and affine regions is given in [309]. A different approach for geometric normalization was proposed in [38] for regions of connected interest points. Depending on the number of points orientation invariance (2 points), affine invariance (3 points), and invariance against homography transformations (4 points) can be achieved thereby. Moreover, [215] proposes a technique to boost each region detector to fully affine invariance by synthetic image transforms of the training objects.

The number of selected interest regions affects the computational complexity of the recognition process, and thus it is useful to filter weak regions out that are probably unimportant for object recognition. [269] proposes a filter that follows the observation that some features are very common while others are rare, and that rare features are more important for object recognition than common ones. The same work proposes a temporal filter for video object recognition that uses only those features that survive for a certain number of video frames. Similarly, [314] uses motion segmentation as a filter for object recognition to reject regions that are situated on

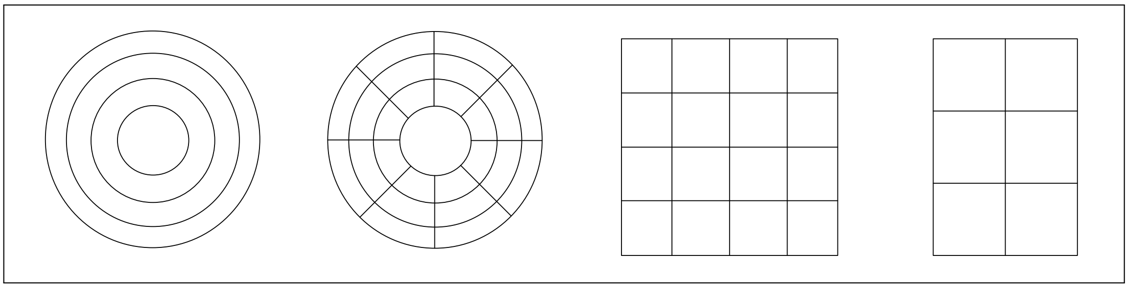


Figure 2.8: Typical region shapes including their sub-regions that are used for descriptor generation. Spin images, log-polar grid, square SIFT grid, and rectangular HOG grid (left to right).

the background layer. Moreover, [191] rejects regions that are centered around interest points in still images with a large principle curvature in only one direction because such edge-points tend to be unstable.

2.3.3 Region Descriptors

After some interest regions have been selected and normalized, region descriptors can be generated to capture either the color, texture, shape, or motion of an interest region. From an information theoretic view, descriptors are vectors with a fixed dimensionality that are compared against each other in the recognition process. The dimensionality of these vectors has a direct influence on the computational complexity and the memory consumption of their recognition systems. Thus, ideal descriptors should be low-dimensional but still able to distinguish between identical, similar, and different interest regions under the transformations described in Section 2.2.1.

A simple solution to extract fixed-length descriptors is it to resize all input regions to the same size and shape (see Figure 2.7), and to store the *pixel values* of each resampled region in a vector [211]. However, such descriptors are very intolerant to object transformations, even when regions are accurately detected and normalized. Instead of using the original pixels, several region transformation and filtering steps are therefore applied to generate more abstract representations of the region content. For instance, the used color models and color conversions are appropriately chosen to achieve illumination invariance [92] while edge images are used to capture the shape and contours of an object. Local binary patterns [4], eigen-images [294], and Kalman filtering [82] are used to build specific descriptors for face recognition whereas the Fourier transform and the Discrete Cosine transform are used as foundation for different MPEG-7 color descriptors [199]. Moreover, *filter responses* of Gabor wavelets [99], the Radon transform [293], and Markov random fields [81] are commonly used. Many descriptors further build on the mathematical approach that any probability distribution is uniquely characterized by its *moments* [297]. Moments are projections of an image function into the space of polynomials, such as the area of an interest region, its average intensity, and its center of gravity. Hu [133] introduced moment invariants in the 60's to the area of pattern recognition and many extensions have been proposed since then to achieve invariance against contrast and illumination changes, geometric transforms, and the rotation of an object [95].

Color histograms were introduced in the early 90's [292] and they present an important step in the evolution of region descriptors as most modern descriptors are based on histograms. The original color histograms gather the amount of quantized colors in a region without information about the spatial color distribution, which makes them rotation invariant. However, spatial information can be added by partitioning the input region into sub-regions with individual histograms and by a concatenation of these histograms. Depending on the used region shape, sub-regions can be differently selected, as shown in Figure 2.8. Spin images [148] use rings of increasing size around the region center to produce rotation invariant descriptors. On the contrary, log-polar grids as well as all rectangular and square regions need an orientation normalization according to the dominant orientation of a region to achieve orientation invariance. In the context of color histograms, it has to be mentioned that several extensions of the basic approach have been proposed. On the one hand, color transforms were used to achieve higher robustness to illumination changes, for instance by the use of color constancy models [101]. On the other hand, sophisticated descriptors like receptive fields [270] try to generalize color histograms in a scale and orientation invariant way.

Similar to color histograms, many texture descriptors are composed of histograms. Instead of using color, they capture the distribution of edges or oriented gradients within gray-level regions. The most popular descriptor of this type is the SIFT descriptor (Scale Invariant Feature Transform) [191] that is usually computed from oriented square regions with a 4x4 sub-region grid. The used orientation histograms are composed of 8 bins, and each bin accumulates the magnitudes of all gradients within a range of 45°. Furthermore, SIFT descriptors are normalized to increase the robustness against color and illumination changes. HOG descriptors (Histogram of Oriented Gradients) [59] are very similar, but they are usually constructed from rectangular regions with a 3x2 sub-region grid, as shown on the right side of Figure 2.8. Moreover, [211] proposed a SIFT-like descriptor, called GLOH (Gradient Location Orientation Histogram), where the orientation histograms are situated in a log-polar grid and a PCA (Principal Component Analysis) is used to reduce the dimensionality. SURF descriptors (Speeded-Up Robust Features) [24] and the Fast Approximated SIFT descriptors [116] are optimized for a significantly faster extraction process while PCA-SIFT descriptors [150] offer a lower dimensionality.

The descriptors mentioned above mainly capture the texture that is directly shown in an input region. However, similar descriptors exist that combine color and texture information [268] and that try to capture the boundary shape of an region. Such shapes are composed of interest points or edges that are situated on the internal or external contours of an object. For instance, MPEG-7 EdgeHistograms capture the number of edges in edge histograms of square regions with 4x4 grid whereas shape context descriptors [25] make use of log-polar grids. PHOG descriptors [33] (Pyramid of Histograms of Oriented Histograms) encode the shape information of regions based on HOGs. Other shape descriptors build on the spatial configuration of interest points [25] or a binary representation of interest regions [96]. BoSS descriptors (Boundary Structure Segmentation) [305] and the boundary descriptors of [11] are segmentation specific descriptors that are generated from the contours of object segmentation regions (see Section 2.3.2) and they are used to measure the geometric relations of object boundary edges.

In addition to color, texture, and shape, many descriptors were recently proposed to capture the motion of interest regions. The HOG/HOF descriptor [161] characterizes the local motion and the appearance of a region by a combination of oriented gradients and optical flow. The 3D extensions of 3D-SIFT [272], HOG3D [329], extended SURF [341], and the 3D MPEG-7

EdgeHistogram [295] operate in a similar way. In a first step, these descriptors try to compute the trajectories of an interest region with optical flow algorithms, KLT tracking [195], or SIFT tracking [43]. The length of trajectories is variable, and thus a quantization is performed to generate descriptors with a fixed dimensionality. Moreover, [350] designed a trajectory descriptor to capture long-term motion patterns whereas further motion descriptors capture the movement of different body parts (arms, legs, torso, and head) [306] and the motion context of moving objects [367].

2.4 Matching and Classification

The actual recognition process is either performed by feature matching or by classification of two local feature sets. One set includes the region descriptors of the objects-of-interest, the other set stems from test images or video frames where the objects should be recognized. Each feature is composed of one or more descriptor types (see Section 2.3.3), information about the region (e.g. the middle point, a dominant orientation, scale, and geometric normalization parameters [23]), and additional information to state from which object or object part it stems. Details about feature matching, classification, and the used dissimilarity measures and kernels are given in the following.

2.4.1 Matching Strategies

Matching strategies compare the features of one set against the features of the other set to identify feature pairs that belong to the same object. The comparison of two features is thereby performed with a distance or *dissimilarity measure* (see Section 2.4.3) that fits to the given descriptor type and the current task but they are usually independent of the actual matching strategy. Thus, different strategies and dissimilarity measures can be mixed. From each matching feature pair it is then possible to compute a *hypothesis* that states which object is situated where and at which pose in a test image or frame, see Figure 2.9.

Simple matching strategies select those feature pairs that are below a certain *threshold*. In this process, each feature can match with an arbitrary number of features from the other set. *Nearest neighbor* strategies select for each feature of set A the feature with the smallest distance of set B. In addition, *k-nearest neighbor* strategies select a predefined number of features, and *reciprocal nearest neighbors* [175] contain only those feature pairs that select each other as nearest neighbor when the two feature sets are exchanged. It is further possible to combine the nearest-neighbor approach with thresholding in a way that the nearest neighbors are only selected if they are below a threshold. Moreover, *nearest neighbor distance ratio* approaches [191] compare the distance between the nearest neighbor and the second nearest neighbor in order to match features that are similar to only one feature in the other feature set. In addition to the combination of different strategies for one descriptor type, matching strategies can be variously combined if different descriptor types are given for each local feature. For instance, [261] combines SIFT and color descriptors tightly in the matching process.

In the usual matching approaches both feature sets are unordered and a linear search iterates over all features again and again to find matching feature pairs. Such exhaustive searches lead

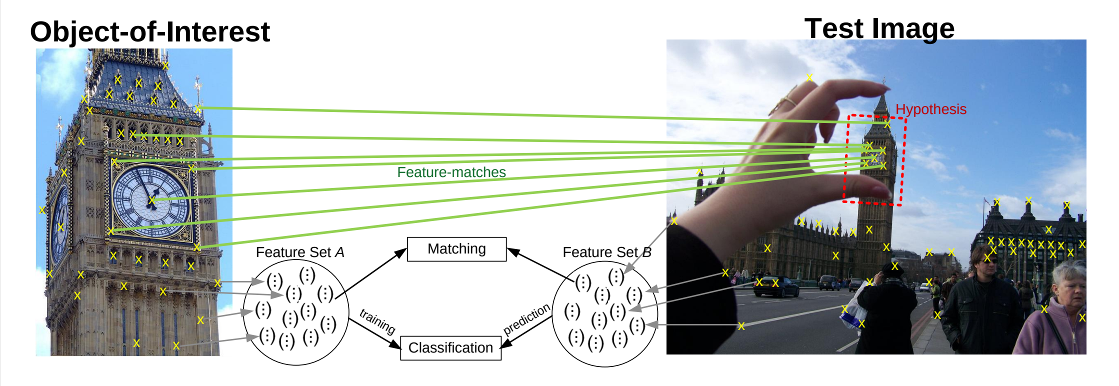


Figure 2.9: Feature matching and classification. First, region descriptors are extracted from interest regions in the image of the object-of-interest and in the test image. Two feature set *A* and *B* are formed from these descriptors. Object recognition is then either performed by direct feature comparison (feature matching) or by classification of the feature sets using a training and a prediction step. A hypothesis of the appearance of the object-of-interest within the test image is generated finally.

to exact results for the price of a high computational complexity that is defined by the amount of features in the two sets, the descriptor dimension, and the used dissimilarity measures. In order to match large feature sets against each other, *approximated matching* strategies are therefore used that achieve speed-ups by several orders of magnitude in exchange for a minor lose of accuracy compared to linear search [219]. These strategies are based on sophisticated data structures, and a certain construction time and memory is required to fill them with task specific data. Kd-trees [98] and different extensions [191, 214] are popular for approximated nearest neighbor strategies. The k-means tree of [229] matches features in the fashion of decision trees using a hierarchical quantization of the region descriptors. Different trees have been further evaluated for local feature matching in terms of accuracy, search time, and construction time [158, 209]. Moreover, it has to be mentioned that most approximated search techniques are adjustable to some parameters like the number of used trees, the tree depth, and the number of nodes. Thus, FLANN [219] offers an automated algorithm and parameter selection that operates on example data and a few information about the desired precision, time, and memory constraints.

After matching, several hypotheses are usually given and combination techniques as well as *verification* techniques are used to decide which hypotheses should be provided as output. The simplest combination is given by *voting* schemes that count how many hypotheses of the same object are given for a test image and the ones with the most correspondences are selected [202]. [191] performs a *statistical verification* by a Bayesian analysis to compute the reliability of the hypotheses that are gathered in one bin of a generalized Hough transform. This verification considers the number of corresponding hypotheses and the assumed object size. More sophisticated *geometric verifications* [269, 271] investigate neighboring hypotheses considering their orientation, size, middle point, and geometric parameters. RANSAC is thereby a commonly used method to estimate the object pose in test images [176, 303], whereas [261] uses the greedy variant. *Region growing* approaches go one step further and try to identify additional hypotheses in the surrounding of existing ones using relaxed matching parameters and newly extracted descriptors. For instance, [89] performs recursive expansions and contradictions to identify more hypotheses and to remove mismatches with local and global filters.

2.4.2 Classification and Machine Learning

Class-level object recognition requires a higher abstraction than individual object recognition, and thus classification techniques are the common method of choice. In contrast to matching strategies, classification techniques do not perform a direct comparison of the extracted features from the two feature sets, see Figure 2.9. Instead, they learn some kind of object models from the object features and predict the presence or absence of these objects in the test set. [176] and [16] use the original region descriptors directly as object model and define local feature matching as classification problem. In the prediction process, they assign an object class label to each feature of the test image. However, in real-world settings similar descriptors are often extracted from different objects which makes direct feature classification difficult. Thus, other approaches first transform the region descriptors of both sets into more generalized higher-level features that are appropriate for modern classification approaches, like support vector machines (SVM) and boosting. A typical requirement of these machine learning techniques is a static descriptor dimensionality with a fixed, non-exchangeable placement and meaning of the descriptor bins. Obviously, this is not given by a simple concatenation of all local features.

The basic approach of local feature-based classification is it to divide the descriptor space into several regions. All descriptors that are situated in the same region belong to the same *visual word*. It is easy to compute a combined descriptor then by counting the number of descriptors for each visual word and by storing these numbers in a simple histogram, called bag-of-features (BoF) descriptor [349]. The most popular space partitioning technique is descriptor quantization by *clustering* with the k-means algorithm, in accordance to text retrieval [269]. Many extensions like approximated and hierarchical k-means clustering [229], as well as similar agglomerative clustering [175], extremely randomized trees [214], and Gaussian mixture models (GMM) [87] have been used for the same task. All visual words together form the *dictionary* or codebook, and the amount of all words defines the dictionary size and the dimensionality of the resulting BoF descriptors. Considerations about good dictionary sizes for different tasks and datasets are made in [319]. In addition to this size, it is another question if a fixed set of primarily collected features is used to learn a dictionary or if an *online learning* is performed where additional features adapt the dictionary from time to time [196]. Certain works further propose the use of object and task specific dictionaries, as described in [242].

The assignment of descriptors to a visual word is then performed by a nearest neighbor matching strategy [311] or by a probabilistic voting [243]. In general, good dictionaries are characterized by the average distribution of assigned features to their visual words [319]. [243] further proposes *soft-assignment* in order to assign each feature to several visual words in the nearby descriptor space with an optional weighting. Recently, *sparse coding* has become a popular alternative to the basic BoF approach. This coding technique approximates feature sets with a linear combination of small dictionaries [355] that are balanced between reconstruction error and reconstruction complexity [26]. Despite their success, it is a common criticism of BoFs and sparse coding that they neglect the spatial order and geometric properties of the original region descriptors. Thus, spatial-pyramids [170] have been proposed as extensions using hierarchical sub-regions in a simple and computationally efficient fashion. Moreover, [55] describes different strategies to incorporate geometric information into BoF models and proposes several weak conditions that are based on the position and dominant orientation of local features. Combined descriptors based on the co-occurrence of visual words and their spatial relations are further used as input for BoF models in [356].

In order to perform the actual classification of test images, the newly generated high-level descriptors are finally fed into a classifier. Although simple classifiers, such as k-nearest neighbor, are possible, SVMs are the most popular choice [87, 320]. In general, SVMs try to separate two classes by a hyperplane and, in the context of object recognition, different kernels have been proposed that are suitable for specific features, see Section 2.4.3. Moreover, extensions like support kernel machines (SKM) were proposed [157] to learn object models with automatically estimated parameters of discriminative classifies and appropriate kernel combinations. Boosting approaches present another popular group of classifiers that are, for instance, used with BoF models in [55]. Implementations, such as AdaBoost that is commonly used for face detection [327], can be used for both, feature selection and object model generation, by a combination of weak learners. Regardless of the used classifiers, a hypothesis states if or, in the case of *regression* methods, to which likeliness (estimated posterior probability) an object-of-interest was recognized in the test image.

Classification techniques can be further distinguished according to the used input data. *Supervised* methods require accurate information about which objects are shown where in the training images. *Weakly-supervised* methods, on the other hand, only require rough object labels whereas *unsupervised* methods use training images without any information about the contained objects. Even though such unsupervised techniques are not suited to learn appropriate object models for most recognition tasks, they are commonly used for dictionary learning [164]. The choice of an appropriate classification technique further depends on the number of training samples that are given from each object-of-interest. Too many samples bear the risk of *overfitting* while too few samples are a knock-out criterion for most machine learning approaches.

2.4.3 Dissimilarity Measures and Kernels

As mentioned before, matching strategies iteratively compare local features against each other to see if they belong to the same object parts. In this process, one has to decide if two features are either similar or dissimilar, and different distance or dissimilarity measures exist for this task. The possible properties of these measures are the following: *identity* means that two identical descriptors have a distance of zero ($d\left(a,a\right) = 0$), and a zero distance further implies that two identical descriptors are given if the *uniqueness* property holds ($d\left(a,b\right) = 0 \Rightarrow a = b$). Non-identical features thus lead to values that are unequal zero and, additionally, above zero if *non-negativity* is given ($d\left(a,b\right) \geq 0$). However, some dissimilarity measures lead to negative values and $d\left(a,b\right)$ does not necessarily lead to the same distance as $d\left(b,a\right)$, which is defined by the *symmetry* property. The *triangle inequality* ($d\left(a,c\right) \leq d\left(a,b\right) + d\left(b,c\right)$) further forms the metric axioms together with uniqueness and identity. Dissimilarity measures have to satisfy these axioms in order to become metrics.

Over the last decades, psychology research developed a number of theories about human perception and the properties of human similarity judgment. Thereby, it was discovered that the triangle inequality does not hold for the human perception and that the symmetry property is not always given [13]. For instance, humans does not necessarily perceive an object shape A equally similar to shape B as B to A [323]. Moreover, humans include semantic information in their similarity estimations and some degree of flexibility is given as it is, for instance, easier to identify similar and dissimilar faces that belong to cultures that someone is in frequently contact with than to unfamiliar ones.

A lot of work has been done to investigate if there is a dependency between the recognition performance and dissimilarity measures, and which dissimilarity measures are best suited for which descriptor type. The Euclidian distance is a popular choice and it is proposed in the original SIFT approach [191] and other works. [189] compares many dissimilarity measures and classifies them into *geometric* measures (e.g. Minkowski family distances, cosine-based dissimilarity, and Canberra metric), *information theoretic* measures (e.g. Jeffrey divergence), and *statistic* measures (e.g. Chi Square statistics). In this evaluation, less popular measures, such as the Canberra metric, perform surprisingly well for content-based image retrieval. In contrast to the mentioned dissimilarity measures, the earth movers distance computes the minimum cost to transform a descriptor a into a descriptor b [263] by a cross-bin movement of descriptor values which makes it only practical for normalized histograms. [240] presents and extension of the earth movers distance that solves this problem, so that it is also suitable for SIFT-like descriptors. In addition to many other works, [323] and [317] present specific distance measures for shape matching and object segmentation.

As described in Section 2.4.2, classification techniques do not perform a direct comparison of the feature sets from the object-of-interest and the training images. Instead, they train object models from the first feature set with a classification method, like SVMs. The high-level descriptors of two or more object classes are thereby used to identify an optimal separation that maximizes the distance between the descriptors of different classes or objects. If the input data is non-separable in linear space, on the one hand, soft margins are introduced to allow a few outliers in the training examples [320]. On the other hand, kernels are used to map the input descriptors onto an alternative higher-dimensional feature space in order to convert non-linear data relations into linear ones. The test descriptors are later mapped into the same space for classification. In this process, the coordinates of mapped descriptors are not explicitly computed but only the inner product of these mappings is given by the kernel function. In opposition to the use of descriptors, distance-based learning uses the distances between these descriptors as input data and the dissimilarity measures described above can be applied, see [121].

Generally, kernel functions are continuous, symmetric, and they process the input data into positive definite kernel matrices. In contrast to the actual classification techniques that are general purpose machines, kernel functions are data and domain dependent [275]. The two most popular kernel families are *polynomial* kernels and *radial basis functions* (RBF). Beside their computational complexity, kernel functions can be distinguished according to the dimensionality of the resulting feature space and their requirements about the input descriptors. Polynomial kernels are finite but they easily lead to a very large feature dimensionality [41], whereas the feature space of RBF kernels (that often use a Gaussian distribution) is possibly infinite. *String kernels* facilitate sub-sequence classification which is not possible with usual kernels that require a fixed dimensionality of their input descriptors. Moreover, kernels are modular, and thus they can be combined to form more complex and stacked kernel functions.

As noted in [41], the best choice of a kernel is still a research question as the classification effectiveness depends on the used kernels, the kernel parameters, and the classification parameters (e.g. soft-margin). Thus, appropriate kernel functions are often selected in a trial-and-error fashion and the parameters are determined by a grid-search approach [132]. For BoF models (see Section 2.4.2) it was empirically found that they perform best with a particular type of non-linear Mercer kernels [275], like the *intersection* kernel and the *Chi-square* kernel [355]. Moreover, [267] explains alternative kernels and how they are commonly used.

2.5 Evaluations and Benchmarks

A large set of evaluation initiatives, dataset, and benchmarks exist that relate to video annotation and retrieval. Their major aim is it to facilitate reproducible research and to make the systems and algorithms comparable. This is similar important for the development of novel algorithms and the improvement of existing ones. In the following, we start with a brief description of the used evaluation methods and metrics before the fundamentals of dataset generation and an extensive overview of available datasets and evaluation competitions are given.

2.5.1 Metrics and Methods

Evaluation methods have to be algorithm-independent, regardless if they are used for video or video scene retrieval, classification, or object recognition. Thus, it is a common approach to compare the generated hypotheses with the ground-truth, and to classify each hypothesis either as *true positive* (tp) if it fits to the ground-truth, or as *false positive* (fp) otherwise. A simple comparison of video, scene, or shot identifiers is often sufficient for video retrieval evaluations. In the case of object recognition, the decision if a hypothesis should be judged as true or false positive is more difficult, as spatial information in form of bounding boxes or arbitrary shaped polygons are given there for both, the hypotheses and the ground-truth. The usual solution to this problem is the computation of mutual region overlaps [85]. An alternative approach is the pixel-level evaluation with statistics about the number of correct and false pixels that is predominantly used to evaluate segmentation approaches. The evaluation of object recognition in video is even more complicated because the temporal consistency of the hypotheses and the ground-truth has to be assured, for instance, by the definition of a minimum number of overlapping frames.In addition to the classification of hypotheses, evaluation methods can further account the number of not recognized examples in the ground-truth. These missing examples are usually called *false negatives* (fn).

Simple evaluation measures are *recall* and *precision*. On the one hand, recall gives the percentage of correctly recognized examples ($r = tp/\left(tp + fn\right)$). On the other hand, precision defines a relation between correct and false recognitions ($p = tp/\left(tp + fp\right)$). Many computer vision algorithms can be optimized to either achieve a high precision or a high recall, and a good trade-off between these extremes is usually desired. *Receiver operating characteristics* (ROC) curves are a graphical representation of recall and precision at different settings. The *area under curve* (AUC) of these curves present a measure that can be easily used to compare different algorithms. Similarly, the *interpolated average precision* (AP), as defined in [85], summarizes the shape of ROC curves using the mean precision at a set of x equally spaced intervals. An alternative to ROC curves are *Detection Error Trade-off* (DET) graphs that show the missed detection rate instead of the recall, see [256] for an example. Curve-based measures are especially useful to compare systems and algorithms that return many results as ranked list, according to the confidence of their hypotheses. These measures are used by the most important evaluation benchmarks that are explained in Section 2.5.3. Further evaluation measures exist that combine recall and precision, such as the *F-measure* [259] that computes the harmonic mean between both measures ($F = 2 * (p * r) \,/\, (p + r)$).

In addition to the accuracy of algorithms, their computational performance (run-time) and resource consumption (memory and hardware requirements) are often considered in an evaluation. Moreover, user studies are sometimes performed as subjective measure, for instance, by accounting the number of satisfying results of interactive retrieval tasks. Thereby, interviews are made and statistical evaluation methods are used to interpret the results [151]. User studies are especially important for video annotation and retrieval systems that heavily rely on user interaction. However, only a few systems provide such studies [51, 183, 338] whereas no extensive user survey of different video annotation and retrieval systems with a sufficiently large number of participants is known to the author.

2.5.2 Dataset Generation

Ground-truth generation is a special case of image and video annotation (see Section 2.1.2) that is meant to develop, improve, and compare algorithms. The basic requirements of these datasets are the broad availability of the data, of baseline results, and of evaluation guidelines to facilitate the evaluation of different algorithms in a uniform way. In the best case, the actual evaluation is automatically performed on an evaluation server and each algorithm is only tested once instead of optimizing them to the dataset. Further requirements are that the objects-of-interest (or whatever someone wants to evaluate) are given in an appropriate quality and quantity with a good balance between all objects or classes, and that they are accurately annotated.

The first question that arises in the dataset generation is: How to select the images and videos as copyright issues often impede the use of existing content. In particular, it is difficult to use professional content whereas the generation of own content is too expensive and out of the scope of most evaluation initiatives. Thus, it is one solution to annotate professional content (e.g. Hollywood movies) and to solely publish these annotations. Researchers that are interested in evaluation have to purchase these movies on their own, obviously in the exact same version to assure that the annotated time-codes fit to the content. Another trend is the use of content from photo-sharing sites and from video portals, such as Flickr [94] and YouTube. Some of these portals allow content publication under the terms of the Creative Commons license. [251] further states that there is a clear need for (object recognition) datasets with realistic and unrestrictive object occurrences and image conditions because there are many datasets that show all objects in the same conditions, at full size, on simple and uniform backgrounds, and without occlusions. Such datasets are not suited to compare modern recognition approaches as a high performance on them does not indicate a similar performance in real-world tasks. The scale of datasets is another issue. Good datasets contain a sufficiently large number of images and videos, and for many tasks the current trend goes to web-scale sized datasets [163]. In addition to content that shows the objects-of-interest, it can be further practical to contain content where these objects are certainly not shown because many machine learning approaches need to learn a default non-object class.

Finally, high-quality annotations need to be correct, accurate, and exhaustive in a way that all occurrences of the objects-of-interest are annotated [85]. Important issues are: what to annotate and in which way. In general, it is not a good idea to let the annotators decide if they, for instance, annotate an entire person or only its head. Thus, detailed annotation guidelines are required to assure high-quality annotations, see [344] for an example. In these guidelines, it is specified which objects should be annotated and which conditions they have to fulfill in terms of visibility, the minimal object size, and the annotator's confidence. Moreover, the roughness of object boundaries and the attributes-of-interest can be specified. In some datasets it might be important to annotate the viewpoint of each object and the assumed recognition difficulty that results from object occlusions or because objects are shown behind glass, in a mirror, in pictures or screens. Moreover, the annotation format has to be defined and the use of common vocabularies eases semantic interpretation issues. In order to achieve such annotations, different ground-truth authoring tools are used, such as LableMe Video [361] and Viper [208] that are described in Section 2.1.2.

2.5.3 Datasets and Competitions

Only a few evaluation sets and benchmarks exist that are specifically designed for video analysis. The TREC Video Retrieval Evaluation (*TRECVID*) that is operated by the National Institute of Standards and Technology (NIST) is the leading initiative for video annotation and retrieval evaluations amongst them [234]. This benchmark is annually performed with different retrieval tasks, such as content-based copy detection, instance search, known item search, semantic indexing, and surveillance event detection. Interactive video retrieval is performed in one of these tasks (known item search) starting from text queries, while other tasks focus on the retrieval of individual objects (instance search), class-level objects, and object related events (semantic indexing). Another annually performed evaluation challenge is *MediaEval* [165] that includes tasks for tagging and geo-tagging of videos whereas different tasks are always raised for the *ACM Multimedia Grand Challenge* [221] by industrial leaders like Google, Yahoo, HP, Huewei, and Technicolor. For instance, in 2009 the Google Challenge [351] aimed at the categorization of web videos with a combination of text and social features. Further interactive video retrieval challenges, such as the *Video Olympics* [287] and the *Video Browser Showdown* [360], are regularly organized as part of international multimedia conferences. In accordance to video retrieval, the annually performed *ImageCLEF* challenge is the most important image retrieval initiative that organizes a couple of different retrieval tasks every year.

The *LableMe Video* [361] database is one of the largest freely available video datasets that already consists of more than 200 different object classes and 70 action classes that are annotated in about 2000 video sequences. Similar to its still image counterpart LableMe [304], this dataset is continuously growing as it is coupled to an online annotation tool. The *Kodak's consumer video benchmark* [221] present a large number of concepts that are annotated in user-generated videos whereas each pixel is assigned to one of 32 semantic objects in the *ComVid* dataset [37]. As the automatic analysis of surveillance videos presents an important branch of research, a lot of datasets are given there. The *PETS* dataset presents an evaluation benchmark for surveillance related tasks, such as person counting, tracking, left luggage detection, and loitering [179]. Similarly, *i-Lids* [138] includes surveillance videos with different event detection and tracking detection scenarios. Systems that successfully participate on this benchmark of the UK government can obtain a security analysis certificate. Further surveillance datasets are provided by the *ViSOR* project [324] and *CAVIAR* [46] that includes alternative events, such as two persons meet each other and window shopping. Datasets that are particularly designed for tracking are mostly given for moving pedestrians and vehicles, see [83,241,333]. Moreover, a lot datasets for human action recognition have been released using Hollywood movies [162], broadcast sport videos, and user-generated YouTube videos, as described in [3, 252].

According to [251], publicly available image collections, such as UIUC [2] and the Caltech datasets [117], have played a key role in the object recognition research although the mentioned datasets are not realistic enough for modern recognition systems. Thus, the more challenging dataset of [89] is often used in the context of individual object recognition. In addition, the *PascalVOC* (Visual Object Classes) challenge currently provides the standard dataset for class level object recognition. This object detection, localization, and segmentation challenge is annually organized and it operates on real-world Flickr images of different vehicles, animals, household objects, and persons. Since 2011, a large scale visual recognition challenge [68, 163] is further held in conjunction to the PascalVOC workshop. A lot of initiatives and datasets are further given for face and gesture recognition as well as for biometrics, like iris and fingerprint recognition. Similar to TRECVID, the *Face Recognition Vendor Test* (FRVT) [100, 246] is regularly performed by NIST to evaluate the state-of-the-art in different face recognition tasks, such as gender, age, and pose estimation. The *FERET* database [245] that is used in these challenges presents one of many face libraries. The *faces in the wild* database [136] and *YouTube faces* database [346] are further examples that provide faces under natural conditions. Furthermore, urban scenes [65], buildings [83], and flowers [228] are contained in some datasets. Evaluation competitions and datasets are also presented for low-level algorithms, such as interest point and interest region detectors [212], and region descriptors [211]. Moreover, a lot of evaluation surveys have been proposed to compare existing dissimilarity measures [189], spatio-temporal features [329], color descriptors [268], and image segmentation systems [365].

2.6 Development Tools

Generally, a couple of different engineering skills are required to generate efficient computer vision algorithms and to integrate them into various programs. In this section, we describe existing tools for the development of recognition approaches and applications that build on object recognition. Such tools facilitate prototyping, the integration and encapsulation of algorithms, as well as feature and approach selection.

2.6.1 Recognition Infrastructures

Computer engineers that want to develop new computer vision algorithms or to integrate existing ones into their applications are confronted with the question which libraries and code snippets they can reuse and which algorithms they have to implement from scratch. A lot of tools, infrastructures, and algorithms are already available starting from video acquisition and decoding to the combination of different features for high-level vision tasks. First, the available tools differ in the used computer languages and their fields of application. Matlab is often used for prototype development whereas C and C++ are the predominantly used languages for industrial applications. The internal image format of the tools also affects the performance because these formats have to provide images in an uncompressed form that is suitable for image processing and that can be used as input and output parameters of different tools and algorithms. In addition to the computer language and internal image format of different tools, the support of different platforms, processor and GPU environments, and the available licenses have to be considered. Furthermore, it might be important if the code of a library is given as open source or if it only provides an API. This is particularly important if somebody wants to modify the given algorithms. Moreover, experimental tools exist that lack of high performance and robustness as they are developed and tested on a single machine, for a single application, and without the aim to achieve industrial applicability.

Popular image and video acquisition libraries that support many different formats (see Section 2.1.1) are FFMpeg [91], libVLC [181], and ImageMagick [139]. Intel's image processing library (IPP) [141] is the favorite choice for internal image representation in C/C++ while no additional library is needed for this purpose in Matlab. Moreover, sophisticated computer vision and machine learning toolboxes exist. The popular OpenCV library [34] provides many state-of-the-art features for C/C++ including Harris corners, Canny edge detectors, SURF descriptors, Viola-Jones face detection [327], HOG body detection, as well as several machine learning approaches. Similarly, Matlab's Computer Vision System Toolbox [53, 204] includes a large set of features and algorithms for Matlab. Other toolboxes are VLFeat [322], OpenCV extensions like [99], Groupsac [118], openSURF [113], and the Torch libraries of [301, 302]. Moreover, some research articles are accompanied by the code or executables of the proposed algorithms in order to facilitate reproducible research, as it is done in [161, 191]. All of the mentioned toolboxes and algorithms provide computer vision functions that facilitate the development of recognition applications significantly. However, a fairly advanced level of software development skills and an in-depth understanding of image processing and computer vision techniques are still required.

In contrast, vision prototyping tools like Papier-Mache [154] and Eyepatch [205] try to simplify the development of recognition applications for developers with no computer vision background. They enable users to create, test, and refine recognition strategies with a visual example-based approach in order to test the capabilities of current recognition approaches. However, the development of applications with vision prototyping tools is restricted to a limited spectrum of tasks, like the detection of certain events from a static camera [154]. On the other hand, most visual prototyping tools are not suitable to develop entire recognition applications at all. They lack of extension possibilities and do not provide the required flexibility to achieve the same results that experienced computer vision engineers achieve with the toolboxes described above [205]. In the context of image retrieval, a Matlab-like programming interface, called RetrievalLab [230], was further presented that contains various features, dissimilarity measures, and classifiers, that is user extendable and includes visualization tools. Moreover, the component-based MetaXa framework [29] was presented for image annotation tasks with expandable feature extraction and metadata enhancement components.

The development of applications that build on object recognition is a particularly demanding task. As mentioned above, several tools exist that reduce the development effort for these applications compared to development from scratch. However, the needed effort and the required engineering skills are still considerable and more advanced higher-level infrastructures are required that integrate various algorithms and that provide simple APIs. [220] recently stated that many recognition architectures are given in the literature but only a few of them present really generalized infrastructures. These works usually propose one specific approach for the recognition of all kinds of objects. In contrast, the authors of the same work presented REIN, a recognition infrastructure for robot vision applications that allows a dynamic configuration of the used algorithms and approaches in order to adapt the system to a certain recognition task. New algorithms can be integrated using three pre-defined interfaces, and configurations of the entire processing chain can be given. However, these configurations have to contain every processing step and the connections between these steps in a complex XML format. Moreover, REIN is restricted by its operation area for robot applications as it runs on top of the robot operating system ROS.

2.6.2 Recognition Approach Selection

[97] pointed out that the recognition of different object types requires different approaches in order to achieve the best results, and [347] stated almost 20 years ago that the time has come to stop searching for a single method that can solve object recognition and to combine different methods instead. As explained in the preceding sections, many different features as well as matching and classification techniques exist, and there are various ways to adjust and combine them. However, the manual process of choosing appropriate algorithms and tuning them for a given recognition task is often performed in a trial-and-error fashion where different approaches and parameter settings are executed on small datasets and their results are manually compared. More advanced selection strategies operate on annotated datasets and use automatic evaluation strategies [16]. The dataset is thereby separated into a training and validation set for approach selection whereas the actual evaluations are latter performed on a different set, see [85]. According to [288], such data-driven approaches increase the probability to learn invariance accurately for specific applications, but require annotated data that is either human labeled [73, 304], synthetically generated by artificial image transformations [144], or semi-automatically gathered by optical flow tracking [212, 288].

The survey paper of [120] presents a general overview of the feature selection problem for classification and includes a check list that summarizes the practical use of feature selection in a compact form. Moreover, feature interdependences, subset selection, dimensionality reduction, and validation are extensively discussed in this work. Feature subset selection differs from recognition approach selection as it tries to select the most discriminative but non-redundant features instead of selecting the best feature types. However, this task is similarly important because irrelevant features might hurt the classification quality and fewer features automatically improve the speed and computational costs of the classification process. In the context of object recognition, [224] selects informative object features for BoF-based object classification using a sparse PCA. Similarly, [76] selects object-part classifiers with an SVM that presents specific object classes appropriately. Other approaches perform subset selection based on the feature likelihood, in accordance to term frequency - inverse document frequency (tf-idf) methods [269]. Boosting approaches, such as the Viola-Jones face detector [327], additionally combine feature selection and localization in a simultaneous process.

In addition to feature subset selection, a couple of works investigate the automatic selection and customization of recognition approaches from different directions: Attempts to optimize the parameters of specific visual features, like SIFT and HOG, are given in [288,342,343]. [321] proposes a kernel-learning approach to select the best feature combination for a task using an SVM framework that works with all types of features. [144] uses a convolutional neural network to learn new features for each task instead of using manually designed, hand-crafted features while [308] learns a couple of interest point detectors automatically. [16] further presents a trainable local feature matching approach that uses a boosting framework. Another feature learning approach for class-level object recognition builds more abstract features with multiple layers of feature hierarchies, as done in the biological inspired work of [223] and in the sparse model approach of [339].

2.7 Summary

In this chapter, we first draw the big picture of current video annotation and retrieval systems from a high-level perspective. The focus lies on the purpose and functionality of these systems and not on the used technologies. This survey contributes to multimedia research with a proper overview of the field whereas all existing studies have been restricted to specific aspects. In addition to summing-up a wide area of research, we incorporate video portals, search engines, and annotation systems that are available on the web and discuss the requirements and intentions of different users.

After this, we review the research areas of object recognition, visual features, and of matching and classification. These areas provide the basis for the intelligent video annotation and retrieval techniques of this thesis. In this context, we investigated why object recognition is difficult for automated systems at all and why computer vision algorithms are significantly outperformed by the human visual system for recognition tasks. Research trends and access points for further reading are mentioned wherever possible. We further discuss evaluation strategies, benchmarks and dataset that are commonly used to compare new techniques against each other and to improve existing systems. Finally, development tools for the generation of computer vision prototypes and products are given.

The presented state-of-the-art leaves a lot of space for further advances and research in video annotation and retrieval systems. Open points that are tackled within this thesis include a flexible object annotation framework for video portals, novel visual features to present the object context, and a better representation of the video retrieval results. The selected topics of this chapter are directly related to these points, and thus references to particular parts are given throughout the residual thesis.

CHAPTER 3

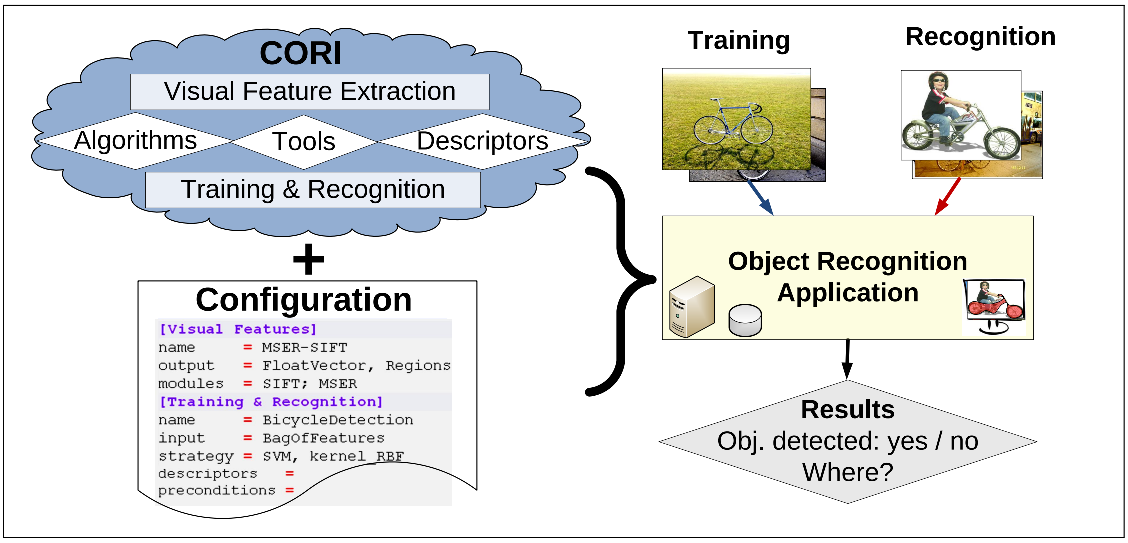
Recognition Framework

In this chapter, we present an object recognition infrastructure that can be adapted to the needs of a broad spectrum of tasks without writing a single line of code. Instead, it can be configured to generate various visual features and to perform training and recognition. The configuration format is quite simple and tools to generate these configurations make it easy to use, even for developers with little computer vision experience. New algorithms and approaches can be added in a reusable way and further tools are provided to cover all parts of object recognition applications from data acquisition over execution and storage to the final result presentation. Moreover, performance considerations have been taken into account to enable the reuse of intermediate results and to support multi-processor architectures. After this, we propose the simultaneous selection and customization of the entire recognition process for specific tasks, domains, or datasets in this chapter. This approach only requires an annotated set of sample images or videos and precisely specified task requirements to select an appropriate setup among thousands of possibilities with iterative analysis strategies.

3.1 Configurable Object Recognition Infrastructure

Several computer vision and machine learning toolboxes [34, 204, 322] exist that are frequently used to develop research prototypes and industrial applications in the field of object recognition, see Section 2.6.1. Although these toolboxes significantly reduce the development effort compared to development from scratch, the needed effort and the required engineering skills are still considerable. Therefore, we propose CORI: a configurable object recognition infrastructure. Figure 3.1 illustrates the use of CORI for a usual object recognition application. This application can be trained to recognize different objects-of-interest. During recognition the system automatically determines if (and where) trained objects are shown in the investigated query images. No algorithm or other code has to be written for the development of this computer vision application because the proposed infrastructure is sufficient for this task.

Generally, state-of-the-art object recognition systems are composed of heterogeneous parts and a set of different skills and expertise are required for their development. For instance, al-



```
[Visual Features]
name      = MSER-SIFT
output    = FloatVector, Regions
modules   = SIFT; MSER
[Training & Recognition]
name      = BicycleDetection
input     = BagOfFeatures
strategy  = SVM, kernel RBF
descriptors   =
preconditions =
```

Figure 3.1: Application development with CORI. CORI consists of algorithms, data types (descriptors), and tools to generate entire object recognition applications that are suitable to train and recognize objects-of-interest with different state-of-the-art approaches. Not a single line of code has to be written therefore because simple configurations are used instead.

gorithmic image processing skills are needed to develop feature extraction and matching approaches while application and database engineering skills are required to store these features efficiently in a database and to present the results adequately. CORI facilitates the separation of different parts right from the beginning and enables the integration of new algorithms and existing computer vision toolboxes in a simple and comfortable way independently of their later use. Changes of a running application can be performed without recompilation and deployment simply by changing its configurations. Considerations about the performance have also been taken into account to ensure efficient applications that support the use of distributed multi-processor architectures. In addition to reduced development and maintenance costs, CORI facilitates rapid prototyping not only for individual object and object class recognition but also for related computer vision tasks like image retrieval, event detection, and motion tracking.

3.2 Usage and Configuration

CORI integrates visual features as well as training and recognition strategies that are suitable for many object recognition tasks. In the following, we describe the functionality of this infrastructure and the needed configuration steps on a cyclist recognition case study. Instead of presenting the experimental results of this case study, several video annotation and retrieval tasks that build on CORI are evaluated later in this thesis.

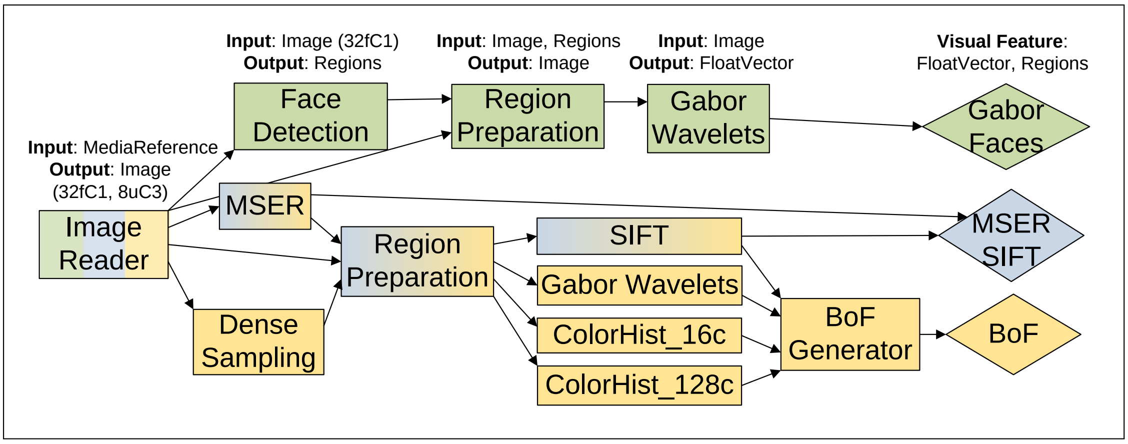


Figure 3.2: Visual feature extraction graph of the case study. Processing steps (labeled rectangles) and their interconnections (arrows) are efficiently ordered to extract the specified visual features of Configuration 3.1 from training and query images. CORI constructs these extraction graphs automatically.

3.2.1 Case Study

We want to recognize specific persons (e.g. professional cyclists) when they are shown in query images together with a bicycle using the following three visual cues or subtasks:

Face recognition: First, Gabor wavelets [99] are used to match frontal face images from a database against those regions from the query images in which a face was detected with the Viola-Jones approach [327].

Bicycle detection: Secondly, we use the popular bag-of-features approach [170] with SIFT descriptors [191], Gabor wavelets and color histograms that are extracted from densely sampled regions and MSER regions [212]. A SVM with RBF-kernel is then trained from images with and without bicycles.

Region matching: As last step, a local feature matching from MSER regions and SIFT descriptors is done according to the individual object recognition approach of [191]. Thereby, we want to find corresponding regions between the query image and the images of the database to recognize objects, like the logo of a bicycle team sponsor.

After these subtasks, a simple fusion is performed that generates a cyclist recognition hypothesis when at least one face was recognized in conjunction with an appropriately detected bicycle or matched region. Although this might not be the most sophisticated cyclist recognition approach, it is well suited to explain the functionality of CORI.

3.2.2 Visual Feature Extraction

Object recognition applications usually start with the extraction of visual features from images or video frames. This process is composed of several steps like image loading and pre-processing (e.g. to scale and normalize the images), region of interest detection and description. The actual steps differ from application to application and Figure 3.2 shows the steps of the cyclist recognition case study. In this figure, steps are shown as labeled rectangles and the arrows be-

```
 1 [Visual Features]  // Region Matching
 2 name    = MSER-SIFT
 3 output  = FloatVector, Regions
 4 modules = SIFT; MSER
 5 ---------------------------------------------------------------------------
 6 [Visual Features]  // Bicycle Detection
 7 name    = BagOfFeatures
 8 output  = IntegerVector
 9 modules = MSER; DenseSampling, scales_3;
10           ColorHistogram, nrColors_16;
11           ColorHistogram, nrColors_128;
12           SIFT; GaborWavelets, filterBiggest_300;
13           BoFGenerator, codebook_cb100.bin
14 ---------------------------------------------------------------------------
15 [Visual Features]  // Face Recognition
16 name                = GaborFaces
17 output              = FloatVector
18 modules_training    = GaborWavelets
19 modules_recognition = FaceDetection; RegionPreparation,
20                       imageSize_32_32; GaborWavelets
```

Configuration 3.1: Visual feature extraction for the subtasks region matching, bicycle detection, and face recognition. The desired recognition modules, their settings, and the output descriptors are adjusted in a simple format.

tween them indicate the interconnections of their input and output. Steps that belong to the face recognition approach are shown in green, to bicycle detection in yellow, and to region matching in blue. The steps that are shown in multiple colors are shared between the corresponding approaches. For the topmost steps (image reader, face detection, region preparation, and Gabor wavelet computation) we further state the IO types. The output of each step (e.g. grey-level images from the image reader) can be used as input for several other processing steps and they can be combined to visual features as shown on the right border of Figure 3.2. The repeated use of the same intermediate results accelerates the feature extraction significantly and reduces the memory requirements.

Significant engineering efforts are required to order the processing steps of visual feature extraction graphs manually. Thus, CORI constructs these graphs automatically from simple configurations, like the one shown in Configuration 3.1. Each subtask of the case study (face recognition, bicycle detection, and region matching) is specified in its own paragraph. The *name* is used to identify the visual features for training and recognition and it can be chosen arbitrarily. The comma separated *output* list indicates the visual feature types and the *modules* string specifies the used processing steps. Thereby, semicolons separate different steps while commas separate a processing step from its parameters, see dense-sampling in Line 9. The first part of a parameter (before the underline) presents its name and all following parts present its values. In addition to the configuration of processing steps, filters can be specified to use only inputs that fulfill certain criteria. For instance, it is possible to use only regions that are above a certain size or to use the 300 biggest regions as specified in Line 12 for Gabor wavelets.

Incomplete configurations also lead to correct feature extraction graphs in CORI. For example, the graph of Figure 3.2 contains an image reader that was not specified in the Configuration 3.1. Furthermore, processing steps can be specified in any order and so region detectors can

```
 1 [Training & Recognition]
 2 name     = BicycleDetection
 3 input    = BagOfFeatures
 4 strategy = SVM, kernel_RBF
 5 --------------------------------------------------------------------------
 6 [Recognition]
 7 name     = FaceRecognition
 8 input    = GaborFaces
 9 strategy = NearestNeighbour && Thresholding, t_0.25
10 distance = EuclidianDistance; JeffreyDivergence
11 --------------------------------------------------------------------------
12 [Recognition]
13 name           = RegionMatching
14 input          = MSER-SIFT
15 descriptors    = FloatVector
16 strategy       = NearestNeighbourDistanceRatio
17 distance       = EuclidianDistance
18 precondition   = FaceRecognition_nrResults >= 1
19 postprocessing = GeometricVerification
20 --------------------------------------------------------------------------
21 [Recognition]
22 name           = Overall
23 input          = RegionMatching, FaceRecognition, BicycleDetection
24 precondition   = FaceRecognition_nrResults >= 1 &&
25                  BicycleDetection_probability > 0.5 ||
26                  RegionMatching_nrMatches > 5
```

Configuration 3.2: Training and recognition strategies the visual features of Configuration 3.1 with machine learning (Line4), combined matching (Line9), and simple matching strategies (Line 16). The last paragraph further specifies a simple fusion of the three subtasks.

also be defined after region descriptors, see Line 4. Feature extraction graphs can further include several instances of the same processing step with different inputs (see Gabor wavelets) or with different parameters, as shown for color histograms with the histogram sizes of 16 and 128 bins, respectively.

3.2.3 Training and Recognition

Most object recognition tasks consist of a training stage that is performed prior to recognition. Visual features are thereby extracted from images or videos that contain objects-of-interest and models are learned from them. These features and models are then persistently stored or temporary loaded during application. The same features are always used for training and recognition of a task but sometimes different processing steps are necessary to generate them. For instance, the face recognition subtask of the case study is trained from frontal face images while recognition is performed on query images where faces can be shown at every position and size. In this example, an additional face detection step is only necessary for recognition. Thus, two separated feature extraction graphs are used in CORI for training and recognition that can be simultaneously configured, as shown in the Lines 18 and 19 of Configuration 3.1.

The actual training and recognition approach of the case study is specified in Configuration 3.2. In the case study, training is performed for the bicycle detection subtask to learn the support vectors. Face recognition and region matching work without additional model learning because they directly compare their visual features from training images and query images in the recognition process. As shown in Configuration 3.2, each recognition subtask is configured in its own paragraph. The *name*, the used *input* features (compare names of Configuration 3.1) and *descriptor* types, the recognition *strategy*, and the used *distance* measures can be specified.

CORI supports three different types of recognition strategies: feature matching, combined matching, and machine learning approaches. Simple matching strategies are nearest-neighbor, k-nearest-neighbor, or thresholding for one input feature type using one distance measure. The Lines 9 and 10 of Configuration 3.2 show the combination of two feature matching strategies using an intersection that is specified with a '&&' in contrast to aggregation ('||'). In this example, faces are recognized when the Jeffrey divergence between a query vector (Gabor wavelet extracted from a face detection region of a query image) and the retrieved nearest neighbor (face from database) is below a threshold of 0.25. Furthermore, it is possible to specify certain *preconditions* that have to be met in order to perform a recognition strategy, as shown in Line 18 where region matching is neglected if no face was recognized. In a similar way, *post-processing* steps can be specified, as shown in Line 19 where a geometric verification clusters individual feature matches to combined hypotheses [191]. As the final result of the case study, a positive cyclist hypothesis is generated when at least one face was recognized in combination with an appropriately detected bicycle or a matched region, specified in the last paragraph.

Please note that we do not present experimental results of the case study because this section proposes a recognition infrastructure and no specific recognition system. Nevertheless, we have made some evaluations to compare the case study results of CORI against the results of a standalone application using the same feature extraction and recognition code snippets without CORI. Thereby, the results of each subtask have been identically from both systems and the computation time of CORI was slightly higher (~1% of the overall time) on a single processor due to the generalization overhead of the proposed infrastructure. However, CORI performed much faster using the build-in multi-processor capabilities (~3 times faster on a quad-core machine).

3.2.4 Tools

CORI contains a couple of tools that facilitate the generation of entire object recognition applications. At first, a configuration generation tool can be used to set up new visual feature extraction graphs as well as training and recognition strategies. Thereby, all integrated modules are presented in a simple user interface and they can be added to specified subtasks. This interface shows the input, output, and a functional description of each module and the parameters and input filters can be set easily.

A second set of tools performs the execution of training and recognition tasks. On the one hand, a graphical user interface and a command line tool are provided to analyze selected files or directories. On the other hand, we provide two watch-dog tools to analyze images on the fly that are transmitted via web-services or inserted in a specified directory. These execution tools facilitate the use of multi-processor machines and distributed architectures. Thereby, it is possible to specify the number of desired threads and processes that should be used to analyze images in parallel to balance the workload. The recognition outputs of these tools are stored in binary files. A simple persistency tool allows the management of these files and the storage of visual features and learned models. Finally, the recognition results can be viewed with a

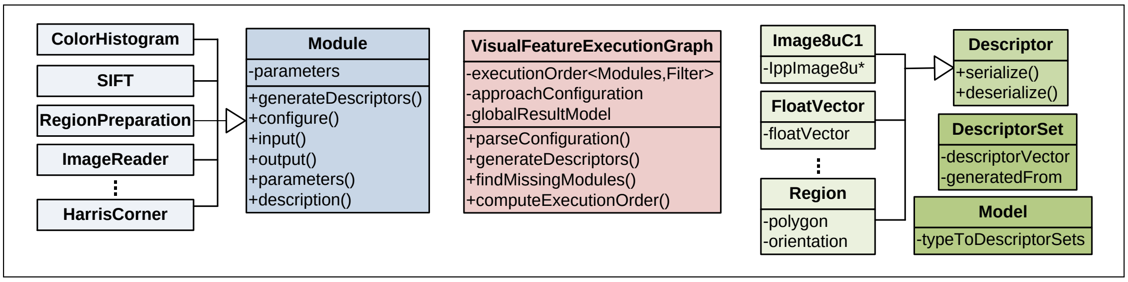


Figure 3.3: Class diagram of the algorithms and descriptors in CORI. Each processing step of the visual feature execution graphs (see Figure 3.2) is implemented in a module sub-class. The functionality lies in the generate-descriptors method. The other methods facilitate their reuse and the automatic construction of execution graphs. Moreover, descriptors are shown on the right figure side that present all input and output data in CORI. Visual feature extraction graphs store all modules and their results.

presentation tool that consists of a GUI where object hypotheses are overlaid on query images and frames.

3.3 Design and Implementation

As shown in Figure 3.1, CORI consists of algorithms, descriptors and tools in its core. Moreover, it provides the functionality to extract visual features and for training and recognition. All parts are written in C++ and the graphical user interfaces are developed with Qt [253]. In the following subsections the design and implementation of these parts are explained in detail using simplified UML class diagrams.

3.3.1 Algorithms

All processing steps and algorithms are capsulated in *module* classes, shown in Figure 3.3. Each module provides methods to explicitly state which *input* descriptors it expects, which *output* descriptors it will generate, and which *parameters* it holds. The internal state of a module instance can be changed using the *configure* method. The actual algorithms and functionality of a module are hidden in the *generate descriptors* method. Each module further contains a *description* method to explain its functionality.

New recognition approaches and algorithms can be added to CORI either as heavy-weight modules (that perform many processing steps internally like the Viola-Jones face detection module shown in Figure 3.2) or as set of light-weight modules that contain smaller processing steps separately. Light-weight modules facilitate the reuse of intermediate results for succeeding modules. This improves computation and memory efficiency. It is further possible to integrate external processes like executable binaries and non-C++ code to CORI. Therefore, a process wrapper exists that executes processes from the command line and that waits until they have finished. This wrapper can be adapted to new processes using appropriate command line strings and customized transformations of their output into appropriate descriptors.

3.3.2 Descriptors

Descriptors present all data of the framework in a type-safe way. They are implemented as smart pointers (Boost library) to prevent unnecessary copy operations. Furthermore, they provide binary *serialization* and *deserialization* capabilities, see Figure 3.3. Descriptors of the same type can be stored in *descriptor sets* together with the information from which modules they stem. Several descriptor sets can be grouped together in a model where they are stored in a map according to their type. *Models* are used as IO-arguments for the generate descriptors method of modules and *visual feature extraction graphs* (shown in Figure 3.3), and they provide methods to add, remove, and access descriptor sets and single descriptors. Most visual features are described by vectors of basic data types, like *float vectors*. More complex features can be stored in user-defined descriptors or as combinations of simple descriptors in a model. User-defined descriptors can be added to CORI by inheritance from the descriptor base class and by overwriting the serialization methods.

In addition to basic data type vectors, *images*, *regions* and *parameters* are the most important descriptors. Furthermore, the recognition results are stored in *hypothesis* descriptors that include the *object-id* of the reference (training) object and the query object. Thereby, it does not matter if these objects are *real world objects*, like bicycles and specific faces, or *media-objects* (images or videos). Moreover, an optional *region* polygon that indicates where the reference object lies in the query object can be given in a hypothesis together with a recognition *probability*.

3.3.3 Visual Feature Extraction

Visual feature extraction graphs are automatically constructed from a given configuration. This process starts with the initialization of a new graph instance, shown in Figure 3.3. These graphs then *parse the configuration* for each subtask separately and perform a validity check. Next, each module specified for a subtask is compared to already existing modules in the graph according to their type, input, and parameter settings. Modules that did not overlap with existing modules are then added to the graph. Furthermore, the specified filters of each module are set in this step. If several filters exist for the same module, an aggregated filter is generated for this module and more specific filters are handed over to the succeeding modules in the execution graph if necessary.

After the modules of all subtasks have been added, it is investigated if they are suitable to generate the specified output by comparison of the module's input and output. Additional modules are necessary if no module is contained in the ordered graph that generates the input of an existing module before it is executed. In order to decide which modules to add, all possible solutions are generated and the one with the lowest complexity is chosen or presented in the configuration tool described in Section 3.2.4. The complexity is thereby computed by the number of modules to add and their estimated computation time. Eventually, the *module-filter vector* is initialized in the execution order and the *subtasks configurations* are computed to store the output type and output modules of each subtask, see right side of Figure 3.3.

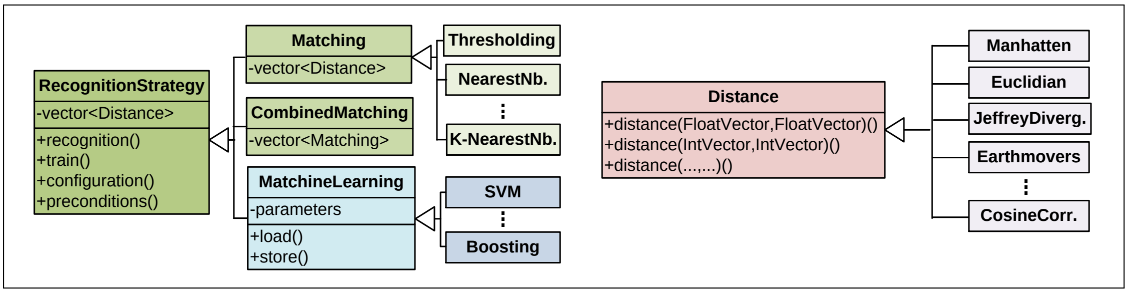


Figure 3.4: Class diagram of recognition strategies and distance measures. CORI supports different feature matching strategies (like thresholding and nearest neighbor search), their combination, and machine learning approaches.

3.3.4 Training and Recognition

During training and recognition all modules of a visual feature extraction graph are executed in the calculated order. Thereby, the required inputs are selected for each module independently from the *global result model* (see Figure 3.3) with respect to the specified input filters. After a module was executed with the generate descriptors method, its output is again stored in this global result model. Finally, the specified visual features of each subtask are selected from the global result model and returned under the given name. These visual features are used to learn new object models for machine learning strategies. However, this model learning step is not necessary for feature matching strategies. Instead, post-processing modules might be used to transform the generated feature matches to an object model, for instance with a geometric verification as proposed in [191].

The base class of all recognition strategies is shown in the top left corner of Figure 3.4. Matching strategies, like thresholding and nearest-neighbor search, are implemented independently of the used distance measures and the same code is used to compute feature matches with different distance measures for different feature types. Furthermore, it is possible to extend CORI by both, new strategies and distance measures without the need to change existing code. The clue behind this functionality is the *distance* class that contains functions to compute the distance between two features of the same type. New distance measures have to be derived from this base class and it is necessary to overwrite the distance functions of those feature types that they support. *Matching* strategies contain one or more distances that are used with the current descriptor type. *Combined matching* strategies (compare the face recognition subtask of Section 3.2.1) contain several matching instances that are executed one after another in the recognition process and combined afterwards using intersection or aggregation.

CORI provides a further layer that wraps the training and recognition process in order to distribute their execution in multiple threads or alternatively in multiple processes that can be executed on different machines. First, the visual feature extraction of several images or video frames is thereby performed simultaneously. Secondly, recognition strategies that operate on single images are executed in parallel for different images.

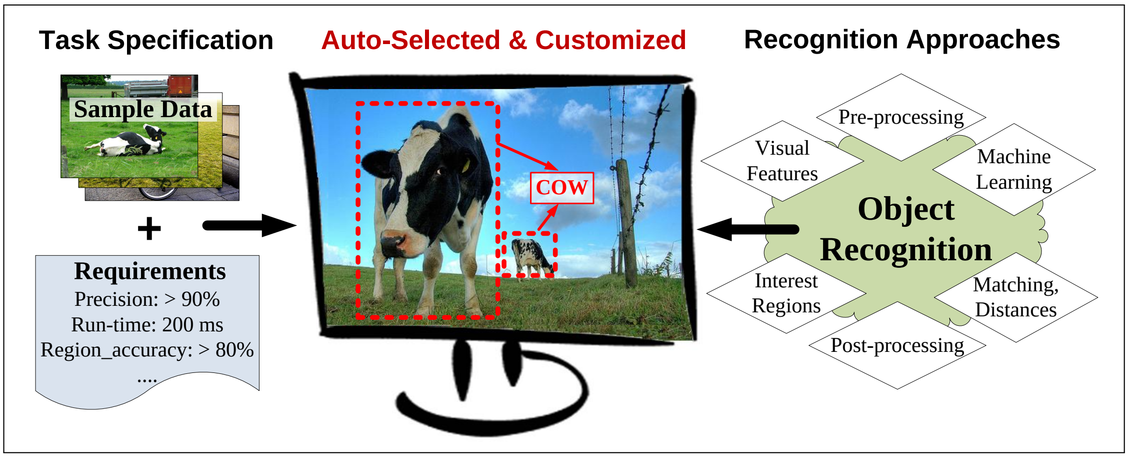


Figure 3.5: Auto-selection and customization of a cow recognition system from precise task specifications (e.g. desired precision and run-time) and annotated sample data. Appropriate object recognition methods that are implemented in CORI are automatically chosen and tuned.

3.4 Recognition Approach Selection and Customization

Humans use their visual system to recognize objects for the interpretation of scenes and images. Computer vision systems try to imitate this process with approaches to recognize individual objects and object classes. Many of these approaches already exist and it requires experience and a fairly advanced level of computer vision skills to select an appropriate approach for an application. Furthermore, recent studies [16, 288, 342] show that the performance of most recognition approaches can be significantly improved when they are adapted to certain tasks, domains, or datasets (see Section 2.6.2). Thus, the research question of this section is: How to automate the selection and customization of recognition approaches for a given task?

This auto-selection and customization depends on both, the task and the investigated recognition approaches. In the example of Figure 3.5, an appropriate approach has to be selected for a cow recognition system that operates quite fast and accurate. On the one hand, recognition tasks mainly differ by the objects-of-interest to recognize and their appearance in the content. Thus, different recognition approaches might be well suited for tasks where all objects are similarly shown from one viewpoint and for tasks with a higher variability. Moreover, requirements about the recognition accuracy, quality, and speed have to be considered in the selection process. On the other hand, recognition approaches mainly differ by the used visual features, matching and machine learning strategies, as well as their settings.

Nowadays, researchers and practitioners usually perform the approach selection and customization in a tedious and time-consuming process by manual evaluation of different setups. In this process, they collect sample data and specify the task requirements before a prototype is developed using the best practice of related tasks. At last, the prototype is adapted to the sample data by replacement of single processing steps or the adjustment of parameter settings. We automate this process by an extensive framework that operates on example-based data and that includes precise task specifications, an efficient recognition infrastructure, and an evalua-

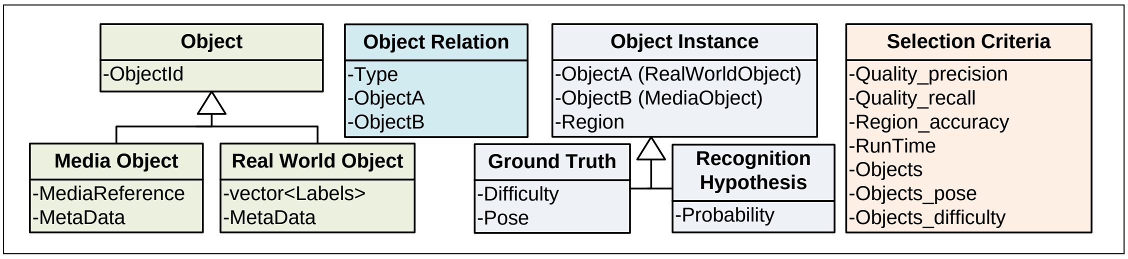


Figure 3.6: Class schema for object annotation, recognition, and task specification

tion tool. The strength of this approach is demonstrated on a case study for face recognition applications in this section and in the video object annotation approach of Section 5.1.

We propose a framework for the automatic selection and customization of object recognition approaches for a given task, domain, or dataset. On the one hand, this framework enforces application engineers to specify their tasks in a precise and machine-readable form. On the other hand, every previously added recognition approach can be investigated for automatic selection and those approaches that offer a certain level of flexibility can be customized as well. In the simplest case, a few parameters are adjusted for a chosen recognition approach, but the selection and customization of the entire processing chain is also possible. In the following, we describe the required task specifications and the recognition setups before details about the auto-selection and customization strategy are provided.

3.4.1 Task Specification

The presented framework operates on example-based specifications of a recognition task where objects-of-interest are annotated in a sample dataset and where the requirements are defined as precisely as possible. These specifications are generated in three steps. First, a dataset has to be collected that represents the task and its objects well with appropriate difficulties and levels of

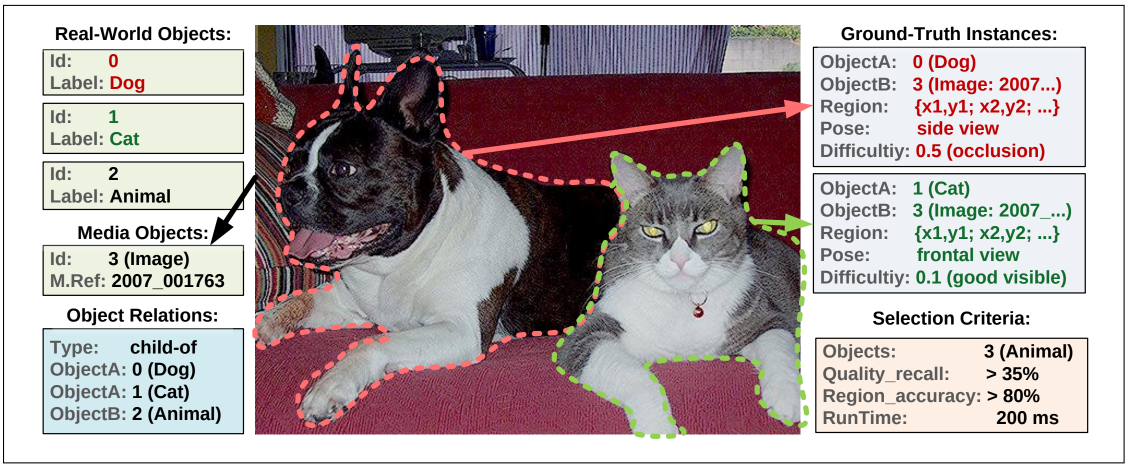


Figure 3.7: Task specification for an animal recognition application. The classes of Figure 3.6 are filled for the annotated animals and the image.

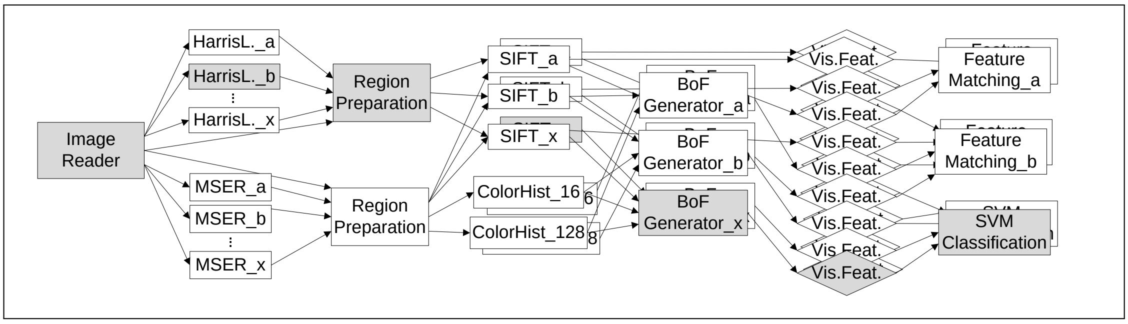


Figure 3.8: Schematic presentation of an analysis graph with a selected setup (shown in gray). Many instances of each component are given in this graph with different parameter settings (indicated by small letters after the underscore).

abstraction. If it is, for instance, the task to recognize all kinds of cars from different views, then the dataset should not only include blue SUVs shown from a frontal view. One way to collect a dataset for rapid prototyping is the manual selection of labeled images from Flickr or Google image search. However, preliminary evaluations showed that the best performance is achieved when the sample images or videos are collected from a realistic application environment.

In a second step, the dataset has to be annotated to specify which objects are situated where. One or more free text labels are thereby used to specify the object's identity and an arbitrary shaped polygon specifies the object's region. As described in Section 2.1.2, a couple of tools exist to annotate images and videos in such a way, like LabelMe [304] and Viper [73], and they generate annotations of different (XML-based) formats. We neither propose another annotation tool nor define a new annotation format. Instead, we present the used class schema that can be applied for all kinds of object annotations. As shown in Figure 3.6 and Figure 3.7, this schema contains *media objects* and *real-world objects* that are both derived from the base class *object*. They contain a unique identifier in combination with the media reference and one or more labels, respectively. The labels can be ambiguous as instances of the same object might be annotated differently when no common vocabulary is shared by all annotators. The object relation *same-as* dissolves such ambiguities while the relations *part-of* and *child-of* are used to model hierarchical structures. Annotated object instances use the *ground-truth* class in contrast to *recognition hypotheses* that present the output of recognition systems. Both classes capture the object-of-interest that is shown in a media object as well as its region, difficulty level, pose and recognition probability.

Thirdly, some selection criteria have to be defined to set the desired recognition quality, accuracy, and speed of a task. The recognition quality is defined by the recall and precision that should be achieved. The region accuracy specifies the required polygon overlap of recognition hypotheses and ground-truth objects to generate a positive match. Another selection criterion considers the analysis run-time. Furthermore, it is possible to specify *selection criteria* for certain objects and their appearance (pose, difficulty, and size).

```
 1 [Visual Features]                     [Recognition]
 2 name    = HarrisLaplace-SIFT          name     = FeatureMatching
 3 module  = HarrisLaplace,              input    = {HarrisLaplace-SIFT,
 4           scales_{0:+0.1:1};                     MSER-ColorHist}
 5           SIFT,subRegions{'2x2',      strategy = Thresholding,
 6           '3x3','4x4','5x5'}                     t_{0.5:0.01:0.8}
 7 -----------------------------------   distance = {L1,L2,Jeffrey,PsiSquare}
 8 [Visual Features]                     postProc = GeometricVerification
 9 name    = BagOfFeatures               -------------------------------------
10 modules = MSER,                       [Recognition]
11           watershed_{0.5:*1.1:10};    name     = SvmClassification
12           ColorHistogram,             input    = BagOfFeatures
13           nrColors_{16,128}           strategy = SVM, kernel_{'RBF','PsiS.'},
14 -----------------------------------              gamma_{1:0.05:5},
15 [Visual Features]                                c_{1,2,3,4}
16 name    = BagOfFeatuers
17 modules = HarrisLaplace-SIFT;
18           MSER-ColorHist,BoFGenerator,
19           codebook_{'cb100','cb500'}
```

Configuration 3.3: Visual feature extraction (left) and recognition (right) setup. The configuration format is slightly extended compared to Section 3.2 in order to specify large ranges of parameter values efficiently.

3.4.2 Recognition Setup

State-of-the-art object recognition systems are composed of heterogeneous components and they are often tailor-made in order to meet the needs of certain applications. Usually, these systems are deeply integrated to the workflow and processing chains of their applications. As a consequence, it is very difficult to adapt such recognition systems to new tasks, and it is a common practice to develop new recognition systems from scratch instead. For this reason, we propose a novel, holistic approach. Recognition components are thereby developed independently from certain tasks in a reusable way. The configurable object recognition infrastructure CORI handles the integration and execution of selected components for a new task, see Section 3.3. With this infrastructure many different setups can be efficiently executed in parallel. CORI enables changes of a running system (parameters as well as the entire processing chain) without recompilation and deployment, simply by adapting its configuration. We use this capability to compare different recognition setups against each other with a minimal development effort.

In this section, a simple extension of CORI is made to allow the generation of multiple setups from a simple configuration, like the one of Configuration 3.3. On the left side, this configuration specifies two visual feature types (Harris-Laplace points [310] with SIFT descriptors [191], and MSER regions [310] with color histograms) and a combined bag-of-features approach [146]. On the right side, we specify that the first two visual features are used for feature matching with a thresholding strategy while the bag-of-features are classified by SVMs [132]. The configuration format is similar to the original format of CORI, but parameters can be defined as intervals or lists of concrete values. Intervals use a *start value : arithmetic expression : end valu*e syntax (Line 3) and lists are comma separated (Line 4). The corresponding analysis graph of Figure 3.8 shows many instances of each component with different parameter settings and that they are all executed in parallel. The output of a component is used as input for several succeeding

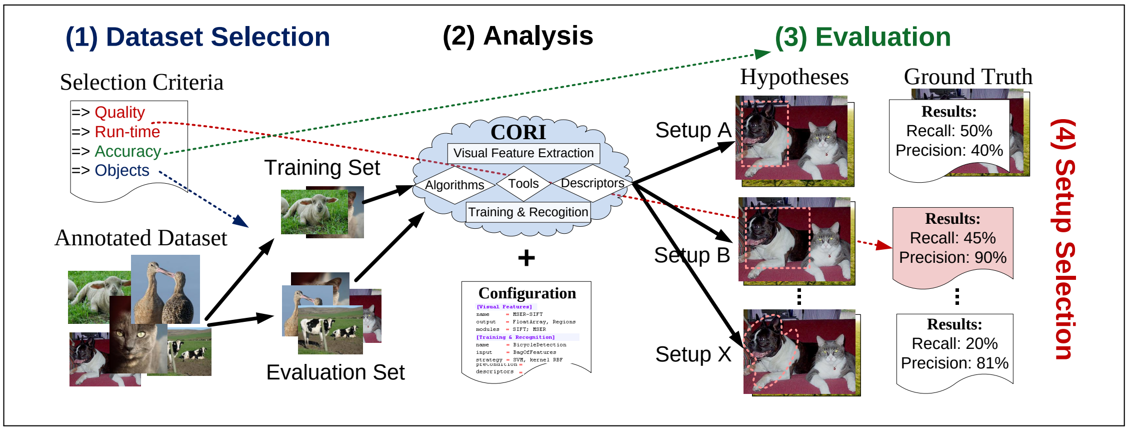


Figure 3.9: Auto-selection and customization process. The four steps dataset selection, analysis, evaluation, and setup selection are executed with the given annotations, task specifications, and recognition setups. The setup that generates the best fitting results (shown in red) is thereby selected.

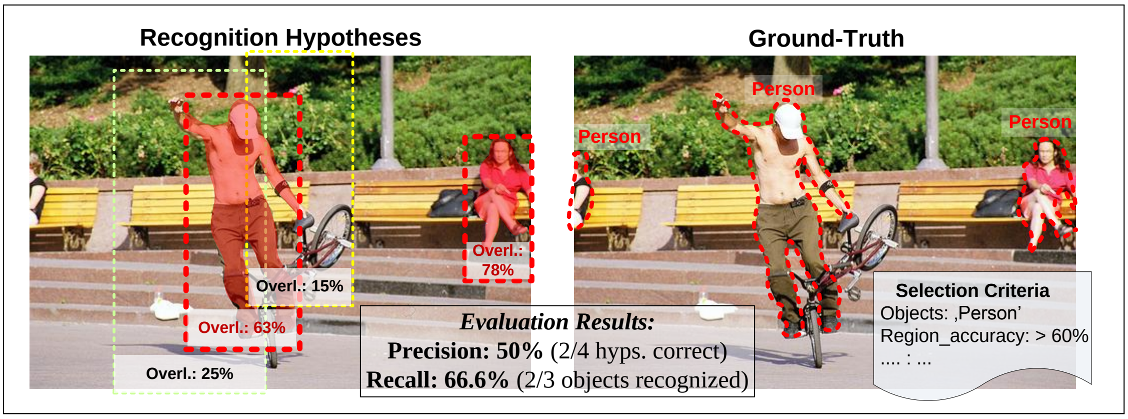
components instead of executing these components again and again in separated graphs for each setup.

3.4.3 Selection Process

After a task was specified and the recognition setups have been defined, the proposed framework starts to select the best path through the analysis graph. As shown in Figure 3.9, the four steps (1) dataset selection, (2) analysis, (3) evaluation, and (4) setup selection are performed. In this process, the annotated images and videos are used to compare different recognition setups against each other and to select the one that fits best to the specified requirements.

Data Selection: In the first step, two sets are selected from the sample data, one for training and one for evaluation. The size and composition of the first set depend on the investigated recognition approaches. On the one hand, machine learning approaches usually train their object classifiers from the same amount of positive and negative examples [132]. Feature matching strategies, on the other hand, can be trained from a few object examples without the need of any negative examples. When the annotated sample data is large enough, an appropriate training set is straight forwardly selected. Otherwise, we support the manual selection of additional examples from Flickr, see Section 4.2.1. The selection of the evaluation dataset is a semi-automatic process that starts with those images from the sample data that are not contained in the training set. Users can then manipulate this initial set to specify the used amount of object instances. In general, the evaluation set should contain the same percentage of object instances as the real application data. For instance, if an object is shown in every 10th image in the application, the same should be true for the evaluation set.

Analysis: During the analysis step, the visual features of each setup are extracted according to the specified recognition setup. CORI makes it possible the extract many different visual features with different parameterizations in parallel and to match them efficiently on distributed multi-processor architectures. However, usually it is not possible to extract all setups in parallel



Evaluation Results:
Precision: 50% (2/4 hyps. correct)
Recall: 66.6% (2/3 objects recognized)

Selection Criteria
Objects: ‚Person'
Region_accuracy: > 60%
.... : ...

Figure 3.10: Evaluation process. The bold bounding boxes in the left image are true positives as their region overlaps with the ground-truth (right image) exceeds the specified region accuracy criterion (box in the lower right corner). Precision and recall are computed straight forwardly, compare Section 2.5.1.

because of memory and run-time issues. For this reason, we start with the analysis of a few sample images. The memory load and the run-time are captured thereby and it is estimated how long a brute force analysis of all setups would take. If the memory load is too high or if the estimated run-time exceeds a few hours, several analysis runs are individually performed. Therefore, different recognition approaches are analyzed separately and a grid search approach [132] is applied to investigate rough parameterization steps prior to finer ones. Moreover, we exclude all configured recognition approaches from analysis that exceed the specified run-time requirements on the first few images.

Evaluation: The evaluation process starts by sorting the generated recognition hypotheses according to their system setups. For each hypothesis of a setup, we then select all ground-truth instances that are annotated in the same image or video frame and that stem from the same object or from an object with an appropriate relation. If no region information is given in the recognition hypothesis, a positive match is generated when at least one of these ground-truth instances exist. Otherwise, all selected ground-truth instances are individually compared against the investigated hypothesis by computation of the region overlap using a polygon intersection. Figure 3.10 shows this evaluation for a task where the ground-truth (right image) has exact object boundaries while the recognition hypotheses (left image) are given as bounding boxes. The specified region accuracy ($> 60\%$) is then used to classify hypotheses as true positives (bold bounding boxes) or as false positives (thin bounding boxes).

Setup selection: Finally, the recognition setup that fits best to the selection criteria is chosen for the application. In this process, we compare the specified recall and precision values against the achieved evaluation results from the investigated recognition setups. If only one measure is specified (recall or precision), we select the setup that meets this condition and achieves the highest value of the other measure. If both measures are specified, we select the setup that meets both conditions and that achieves the highest F-measure. When no setup was found that meets the conditions, we present the setups with the highest recall, the highest precision, and the highest F-measure to the user for manual selection. The specified quality requirements

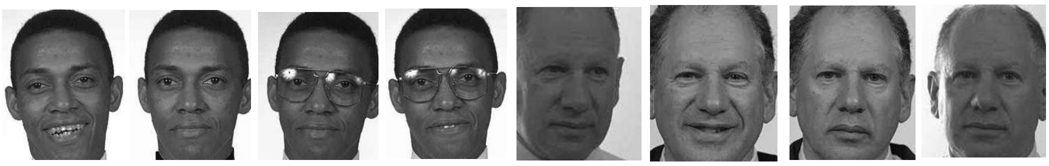


Figure 3.11: Example faces of the FERET dataset.

are only used for the auto-selection and customization process, which means that there is no guarantee that these values are later achieved in the application with real data. However, experiments have shown that similar results are achieved when the correlation between the evaluation dataset and real data is high.

3.4.4 Experiments

We demonstrate the capabilities of the proposed auto-selection and customization framework for a couple of different applications in the area of face recognition from a single image per person. In these applications, it has to be decided which trained person is shown in a query image. As pointed out in [294], this task is non-trivial and different recognition approaches (including global features, local features, different matching strategies, and different settings) might achieve good results. Face recognition is usually performed on image regions that have been identified by face detection approaches like the popular Viola-Jones AdaBoosting [327]. For the sake of simplicity, we use the FERET face database [247] to avoid annotation issues and to make the experiments comparable to other works. However, exactly the same experiments can be done for other tasks like face recognition in surveillance videos.

Dataset: The used face database [247] consists of 1702 gray-level images of 256 different persons. From each person at least 4 different portray photos are given with different facial expressions, with and without glasses, from slightly different views, and with different lighting conditions, see Figure 3.11. In the experiments of this work, we divide these images into three datasets. The first one is the training set. It consists of 128 randomly selected images from different persons. The remaining images are divided into two equally large sets for auto-selection and customization, on the one hand, and to test the performance of the selected approaches, on the other hand. These sets consist of 355 positive examples (trained persons) and 432 negative examples (persons that are not trained).

Task and recognition setup: We use following selection criteria to evaluate different task requirements. (1) Recall > 90%, (2) precision > 90%, and (3) best F-measure whereby one evaluation was done without run-time requirements and another one with the requirement that recognition should not exceed 200 ms per query image. The high-precision scenarios assume that a trained person is only returned when the system is quite sure that the query image belongs to this person. In the high-recall scenarios, more than one person is usually returned for each query image. This might be appropriate for applications where the user makes the final decision.

We employ a set of visual features that are globally extracted from the face images and locally extracted around interest regions. Each feature is used for different matching strategies

Type	**Name**
Visual Features	SIFT, Gabor wavelets, MPEG-7 ColorLayout
Interest Regions	dense sampling, DoG, MSER
Pre-Processing	color normalization, image scaling
Filtering	high-contrast, minimum region dize
Matching Strategy	nearest neighbor, k-nn, nn-distance ratio, tresholding
Dissimilarity Measures	L1, L2, Canberra Metric, Jeffrey divergence, Psi Square

Table 3.1: Recognition components that are used for auto-selection in the experiments.

with several dissimilarity measures. Furthermore, optional pre-processing, post-processing, and filtering steps are investigated. Table 3.1 gives an overview of the used components, a description of them can be found in Section 2.3 and Section 2.4. Each component has one or more parameters for customization.

Results: Table 3.2 shows the achieved results for the test set. Note that the set was not used for auto-selection and customization, and thus some results are below the specified selection criteria. The numbers in brackets give the differences to the results of the selection set. For all three scenarios without run-time requirements (left columns), local SIFT features have been selected from dense sampled interest regions. However, the parameterization of each component (e.g. the used sampling scales and step sizes) as well as the matching strategies and dissimilarity measures are different. In the first scenario, features are matched with a nearest neighbor strategy with L1-distance, and a feature voting component generates recognition hypotheses for persons with at least 17 votes. In the second and third scenario, a k-nn feature matching (k = 3 and 5) was used with Euclidian distance combined with an alternative feature voting where the percentage between the highest entry and the second highest entry has to exceed a certain value (22% and 31%). Global Gabor wavelets with a nn-distance ratio strategy have been selected in the second and third scenario with runtime restriction (right columns). Different distance measures are thereby used (Jeffrey divergence and L1). In the high-recall scenario, a local feature DoG-SIFT approach was selected with an image scaling (to 128x128 pixels) and a high-contrast filter.

As shown in the brackets of Table 3.2, similar results have been achieved between the selection set and the test set in all scenarios. Differences of less than 5% are given, and the results slightly shifted to the middle of the trade-off between recall and precision. This indicates that it is possible to select recognition approaches automatically that meet the specified requirements when the sample data is highly correlated to the application data. The achieved results did not

Selection Criterion	***Run-time < ∞***		***Run-time < 200 ms / image***	
	Recall	**Precision**	**Recall**	**Precision**
Recall > 90%	87.3% (- 2.7)	36.4% (+ 0.9)	85.9% (- 4.2)	17.3% (+ 2.1)
Precision > 90%	79.5% (+ 1.1)	90.3% (- 2.5)	63.9% (+ 2.1)	88.6% (- 1.8)
Best F-Measure	68.6% (+ 4.2)	95.1% (- 4.4)	66.1% (- 1.3)	85.2% (- 2.9)

Table 3.2: Results of the test set including the measured differences to the selection set (in brackets).

improve the state-of-the-art for face recognition from a single image per person, but the achieved results are close to the top ranked approaches in [21]. In our experiments, the auto-selection and customization process investigated between 1500 and 5000 recognition setups for each scenario and took between 5 and 12 hours to complete. Further discussions about the run-time and storage requirements of the proposed recognition approach selection are given in 5.3.3.

3.5 Summary

In this chapter, we propose a configurable object recognition infrastructure (CORI) that is suitable to generate different object recognition applications without high development efforts. Therefore, two simple configurations are sufficient that are easy to generate even from developers with little object recognition background. The infrastructure is extendable for all kinds of visual features, training and recognition strategies. Moreover, CORI provides an interface for the use of external executables. High performance is assured by the support of multi-processor architectures and an intelligent reuse of intermediate results within automatically constructed feature extraction graphs. A case study is used to demonstrate the performance, strength, and usability of CORI compared to the conventional use of computer vision toolboxes. In comparison to the most similar recognition infrastructure REIN (see Section 2.6.1), CORI provides a higher degree of freedom in its extension opportunities, a more powerful but easier to generate configuration format, and it runs on general purpose computers.

Moreover, CORI facilitates the selection and customization of recognition approaches for complex task specifications that can hardly be achieved otherwise. The entire recognition process is investigated in this process to select an appropriate setup for a given task, domain, or dataset. In contrast to this holistic approach, related works optimize only specific components of the recognition process. We use different visual features and recognition approaches, as well as different parameter settings of all components to select appropriate approaches. For this purpose, many recognition setups are generated from a simple configuration file and they are executed in parallel. In order to cope with the complexity of thousands of setups, we further propose an iterative analysis strategy. As a proof-of-concept it is shown that the presented approach works efficiently for face recognition from a single image per person. Different recognition approaches are selected thereby for each task requirement and the achieved results are close to the state-of-the-art.CORI and the proposed auto-selection provide the foundation for the developed visual features, video annotation and retrieval techniques that are presented in the following chapters of this thesis.

CHAPTER 4

Novel Visual Features

As mentioned in the last chapter, a lot of state-of-the-art features and object recognition approaches are already integrated in the proposed recognition infrastructure. Motivated by the fact that current visual features mainly account for the shortcoming of automated recognition systems compared to the human visual system [238], we employ the automatic approach selection and customization of Section 3.4 to develop and integrate novel visual features and feature extraction techniques into this infrastructure. These features are especially suited for object recognition and specific video annotation tasks.

Visual features are usually extracted globally from entire images or locally from interest regions. In this chapter, we propose different approaches to extract semi-local features from segmented objects in the context of object recognition. The focus lies on the transformation of arbitrarily shaped object segments to image regions that are suitable for the extraction of features like SIFT, Gabor wavelets, and MPEG-7 color features. In this region transformation step, decisions arise about the used region boundary size and about modifications of the object and its background. Amongst others, we compare uniformly colored, blurred and randomly sampled backgrounds versus simple bounding boxes without object-background modifications. We further present an interest region detector that tries to gather entire objects that are repeatedly shown in different images and video frames. Starting from a textual annotation, this approach can be further used to segment objects-of-interest just by leveraging the knowledge from large-scale collections of annotated images. Moreover, two motion features are given in this chapter that are especially designed for action scene detection in professional and user-generated videos.

4.1 Semi-local Features

Recently, a set of object recognition approaches have been proposed where segmentation is used as a pre-processing step [177, 236, 256, 264]. They outperform sliding window approaches although almost the same features and classification techniques are used. We believe that customized features that are less distracted by the object's background can further improve these

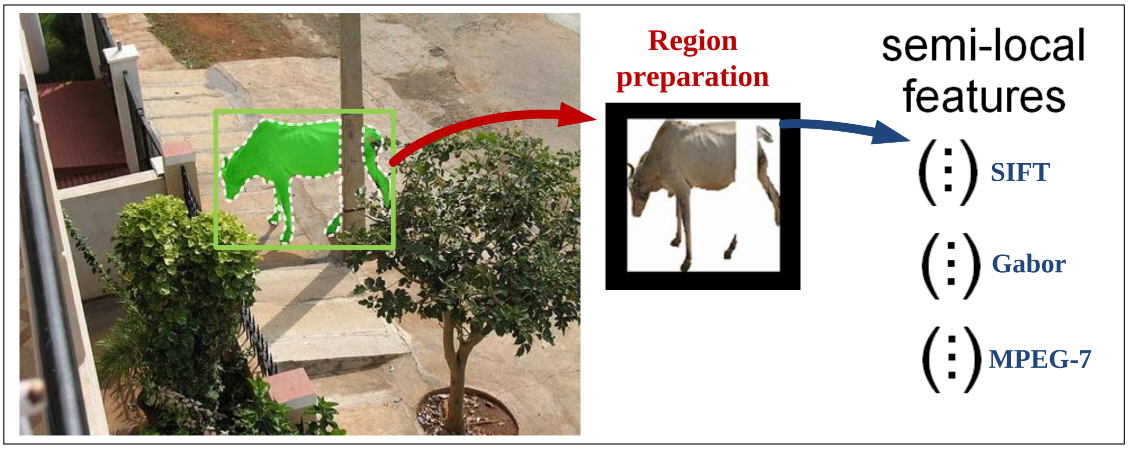


Figure 4.1: Semi-local features. A region that covers the entire segmented cow is prepared (here: background replaced by white pixels) to extract color and texture features from it.

results. Thus, the research question of this section is: How to extract state-of-the-art texture and color features best from segmented objects to improve recognition systems? The proposed semi-local features exploit different region modifications to set the focus on specific object properties. Furthermore, these features are simple and fast to compute which makes them suitable to assist segmentation-based object recognition systems.

Generally, the recognition of class-level objects in real-world images is a challenging task for automated systems that is far from solved, see Section 2.2.1. Objects can be situated everywhere and at every size in an image. They can be occluded and shown under all kind of perspective distortions or under different lighting conditions. Moreover, intra class differences and inter class similarities can complicate this task. Even humans sometimes fail to distinguish between closely related classes, such as bicycles and motorbikes, when only a single image with difficult examples is shown. However, the complexity of object recognition can be reduced when a set of segmented object hypotheses are given in the first place [177], as it is accurately known where to search for an object.

We propose the extraction of well-established image features semi-locally from segmented objects. Thereby, color and texture features are generated from image regions that contain the entire object. We use the term semi-local features because these features are locally extracted from the image but globally extracted from the object. Furthermore, we show that the use of differently prepared image regions facilitates the power of these features. For instance, the object's background is excluded and replaced by white pixels in Figure 4.1. We extract and classify semi-local features from segmented objects in following steps. First, a set of transformed image regions are prepared from every segmented object. Next, different color and texture features are extracted from these regions and stored in a database. The features of each object are then matched against the features of all other objects using a nearest neighbor strategy with several dissimilarity measures. At last, we evaluate the percentage of correctly matched features for each object class.

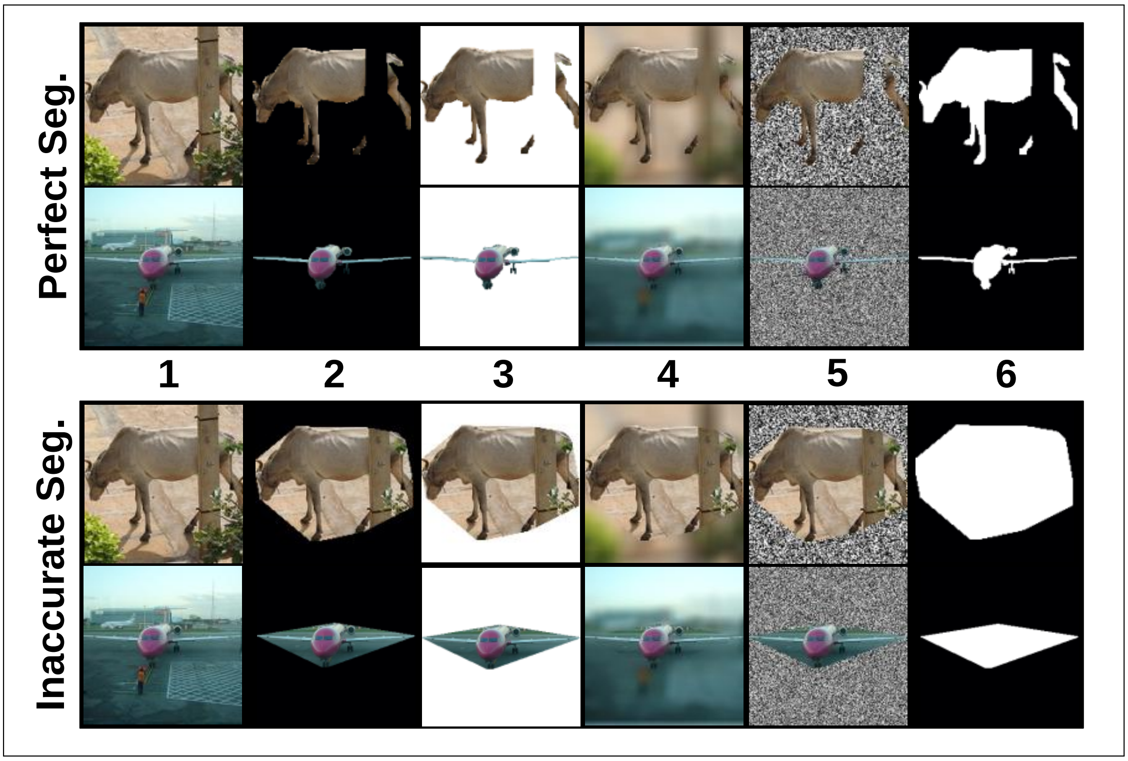


Figure 4.2: Region preparation techniques from perfect (top row) and inaccurate segments (bottom row), shown for the cow of Figure 4.1 and the plane of Figure 4.3. The columns correspond to the regions explained in Table 4.1.

4.1.1 Region Preparation

In the region preparation step, we use different object-background modifications, segmentation accuracies, and bounding boxes to transform object segments into regions for semi-local feature extraction. In the following, these region preparation methods are explained and their effects on the resulting feature properties are discussed.

Object-background modifications: We use six different modification techniques, shown in the columns of Figure 4.2 and in the rows of Table 4.1. Region 1 is equivalent to bounding boxes without segmentation. No focus is set to specific properties of the object in these regions. In the opposite, shape is the only attribute left to describe in Region 6. In Region 2 and Region 3 black and white backgrounds are used. These regions set the focus to the object shape and its content (texture and color). Region 4 keeps the characteristics of the original background although the object is focused and the object boundaries are sharpened. We use Gaussian smoothing to blur the background of these regions heavily. The Gaussian noise of Region 5 also sets focus to the object but with fewer weighting of the object shape. In preliminary experiments, we have tested further object-background modifications (e.g. object boundary expansion) but the six selected ones performed best.

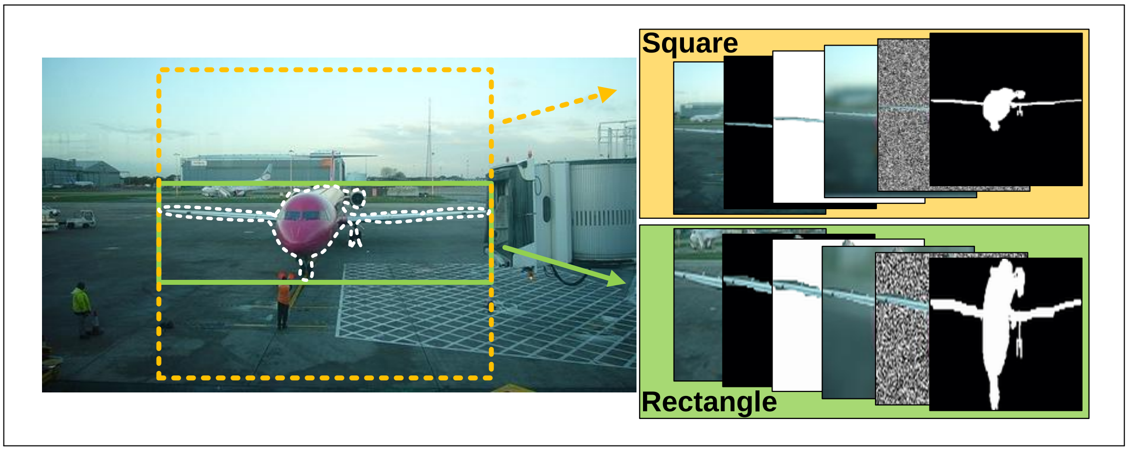


Figure 4.3: Bounding boxes. Two different bounding boxes are used for region preparation from each segmented object (left). The square bounding box includes more background but does not change the aspect ratio of the resulting regions (right).

Segmentation accuracy: As shown in Figure 4.2, we use two different segmentation accuracies. On the one hand, perfect segmentations are given from the Pascal VOC dataset [85]. The object pixels are thereby used as foreground and all others are used as background. On the other hand, we simulate an inaccurate segmentation using the convex hull of all pixels that belong to a perfectly segmented object. No holes are retained in this approach but the actual object shape is heavily changed. In the concluding case study, we further use an automatic object segmentation approach. For these experiments no information about the segmentation accuracy is given.

Bounding boxes: Most visual features are extracted from square image regions. However, segmented objects are given as arbitrarily shaped polygons or image masks, and thus we operate on bounding boxes around such object segments. As shown in Figure 4.3, we select two different bounding boxes for each object. First, we use tight, rectangular bounding boxes that touch the segment bounds on all four sides. These regions are resized to squares in a pre-processing step. Secondly, we use square bounding boxes that touch the object bounds only in the larger dimension. These regions contain larger parts of the object's background but no additional resize step changes the aspect ratio of these regions.

Region	***Object***	***Background***	***Focus***
Region 1	original	original	none
Region 2	original	black	shape & object
Region 3	original	white	shape & object
Region 4	original	blurred	object & background
Region 5	original	Gaussian noise	object
Region 6	white	black	shape

Table 4.1: Object-background modifications. The focus of each region and the properties of the resulting semi-local features reflect these modifications.

4.1.2 Features and Classification

Four popular texture and color features are used in the experiments: SIFT [191], Gabor wavelets [99], MPEG-7 ColorLayout and ScalableColor [199]. In these experiments, we extract only one feature of each type from the entire object region without interest point detection. This is also true for the used SIFT descriptors that gather orientation histograms of 4x4 slightly overlapping subregions. As described in Section 2.3.3, Gabor wavelets are computed with a bank of orientation and scale sensitive Gabor filters, the mean and standard deviation of these filters is used as final feature. ColorLayout presents the spatial distribution of colors by applying a discrete cosine transform on the average pixel values of 8x8 regions. In contrast, ScalableColor features use a quantized HSV color histogram and a discrete Haar transformation to build a scale invariant color feature. We omit to add specific shape features because the used texture features extracted from Region 6 (white object on black background) already present effective shape features.

We compute the nearest neighbor for the segmented objects using all described region preparation techniques and feature types independently. Thereby, each segmented query object is matched against all other segmented objects in the dataset. The object class of the nearest neighbor is then used to determine the class of a query object. We perform this nearest neighbor classification with the dissimilarity measures of Table 4.2. These measures have been chosen according to their high performance for image retrieval with global features in [189]. We believe that more sophisticated classification approaches can be used to achieve better recognition results, but it is out of the scope of this thesis to identify the best classification strategies. Instead, we try to perform a fair comparison between the proposed feature extraction techniques and want to show how these features can be used to improve existing recognition systems.

Measure	***Formula***
Minkowski Family Distances	$\left(\sum\limits_{i=1}^{n} \lvert a_i - b_i \rvert^p\right)^{\frac{1}{p}}, p = \left\{\frac{1}{2}, 1, 2\right\}$
Cosine-Based Dissimilarity	$1 - \frac{\sum\limits_{i=1}^{n} \lvert a_i * b_i \rvert}{\sqrt{\sum\limits_{i=1}^{n} a_i^2} * \sqrt{\sum\limits_{i=1}^{n} b_i^2}}$
Canberra Metric	$\sum\limits_{i=1}^{n} \frac{\lvert a_i - b_i \rvert}{\lvert a_i \rvert + \lvert b_i \rvert}$
Jeffrey Divergence	$\sum\limits_{i=1}^{n} \left(a_i \log \frac{a_i}{m_i} + b_i \log \frac{b_i}{m_i}\right)$
Chi Square Statistics	$\sum\limits_{i=1}^{n} \frac{(a_i - m_i)^2}{m_i}, m_i = \frac{a_i + b_i}{2}$

Table 4.2: Dissimilarity measures used to classify semi-local features: n specifies the dimension of feature a and b.

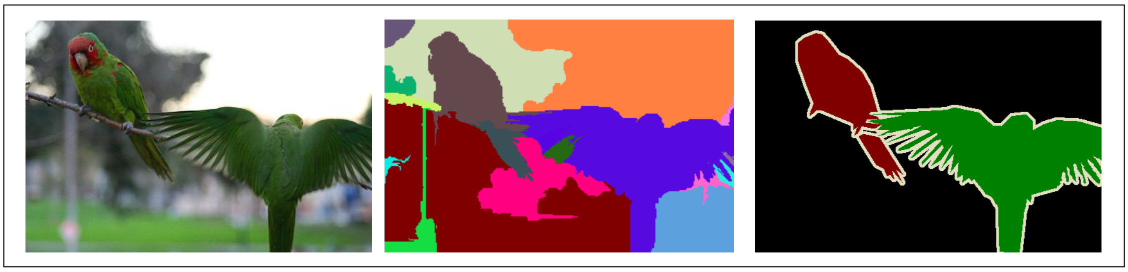


Figure 4.4: Segmentation output: The input image (left) is transformed into a set of image regions (middle) or to segmented objects with black background and white boarder pixels (right).

4.1.3 Implementation

Most object recognition systems consist of heterogeneous components and they are deeply integrated into their application workflow. This makes it difficult to alter specific components of these systems if changes are required for some reason. In contrast to this practice, we use the recognition infrastructure of Chapter 3 (CORI) to enable the interchangeability of different segmentation approaches, region preparation methods, visual features, and dissimilarity measures. On the one hand, CORI facilitates the development of these components in a reusable way, independent from specific tasks. On the other hand, new processing chains can be arranged with simple configurations by the selection of desired components and their parameters.

In this section, we focused on the development of two novel CORI components: a segmentation wrapper and a region preparer. The segmentation wrapper operates on the output images of typical segmentation approaches instead of supporting only one single approach. As shown in Figure 4.4, these output images contain each segmented region in a different color for both, image and object segmentation. The only difference is that object segmentation approaches (right image) only segment those regions that probably belong to an object while non-object pixels are black and object boundary pixels are shown in white. In the experiments, perfect object segmentation images are used and inaccurate segmentations are simulated in a further preprocessing step. However, we also want to support a fully automated object recognition workflow. Thus, we implemented the segmentation wrapper in a way that it is able to execute various segmentation approaches as an external process. Currently, this works for every segmentation approach that is executable from the command line with the arguments *input image directory* and *output directory*. Eventually, the segmentation wrapper returns the bounding box (square or rectangle) and the pixel mask of each segment.

The region preparer uses the original image and the segmentation wrapper results as input in order to generate the proposed image regions for semi-local features generation. In this implementation, we first generate a new image with the size of the bounding box of this segment and fill it with the background of the current region, see Table 4.1. The smoothing of Region 4 is thereby applied by convolution with a Gaussian mask using the Intel Performance Primitives [141]. After this step, each pixel of the new image that is given in the region mask is replaced with the original pixel from the input image or with a white pixel for Region 6, respectively. Finally, we resize the new image to a fixed size of 64x64 pixels for the computation of color and texture features.

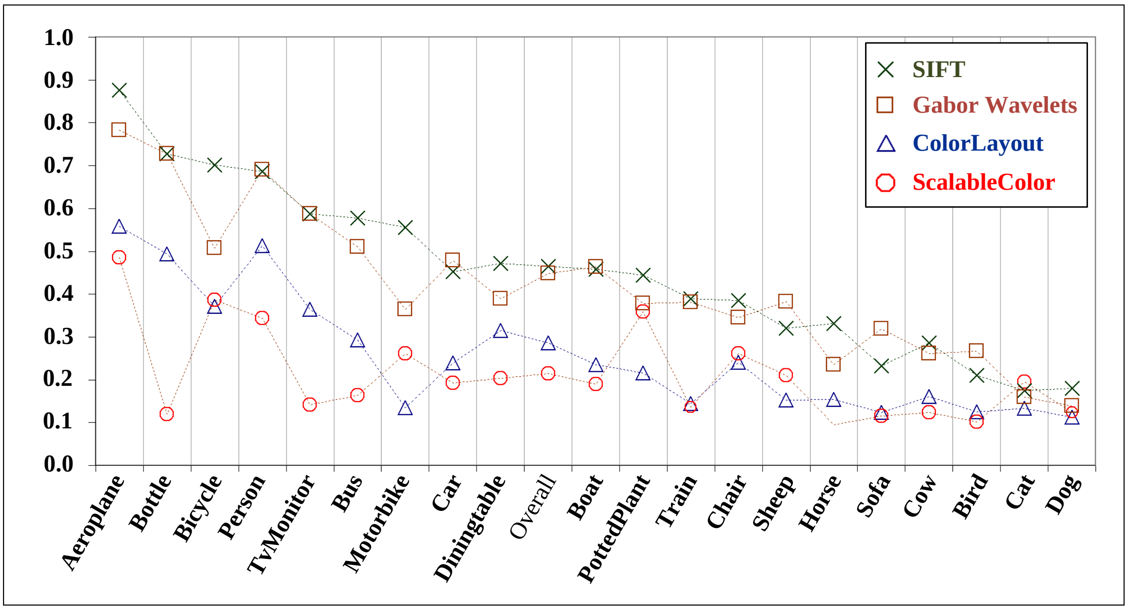


Figure 4.5: Recall per object class from perfectly segmented objects. For each feature type the results of the best region are shown. The object classes are sorted according to their highest result from left to right.

4.1.4 Experiments

In the experiments, two different evaluation strategies are used. On the one hand, we computed the recall of correctly classified objects for each object class and for all classes combined. On the other hand, we did a precision-at-k evaluation to count the number of query objects with at least one correct match in the top k entries (k = 1 to 10). Afterwards, we performed a small case study to investigate semi-local features in combination with automatic image segmentation approaches.

We used the open Pascal VOC 2010 segmentation dataset [85] for all experiments. In this dataset, 20 different object classes (see x-axis of Figure 4.5) are perfectly segmented in 1928 Flickr images. The ground-truth contains a total number of 4203 objects whereby several object classes occur more often than other ones. For instance, 928 persons and 108 dining tables are given. All images are provided with JPEG encoding and a longer dimension side of 500 pixels. The results are organized according to following aspects: the suitability of semi-local features for object detection; the role of region preparation, segmentation accuracy, used image feature types, and dissimilarity measures. Figure 4.5 and Table 4.3 are used to illuminate these points. Both show the achieved recall of a nearest neighbor classification with Jeffrey divergence on squared bounding boxes.

Semi-local features: Figure 4.5 shows that the recall rates of the best matching object classes are significantly above 50% for texture features. Furthermore, the results of all objects are clearly above random classification (5%) independent of the used feature type. The fact that all 4-legged animals (sheep, horse, cow, cat, dog) are below the average, indicates that inter-class similarities decrease their classification. As shown in Table 4.3, the highest overall recall of 46.5% was achieved with SIFT features from perfectly segmented Region 6. Moreover, 80%

of all objects have at least one correct match within the first 10 retrieved objects for the same configuration. These results clearly indicate that semi-local features are able to facilitate the detection of accurately segmented objects.

Region preparation: Table 4.3 shows that texture features achieved the best results on Region 6 (white foreground on black background) where only shape information is given. This is also true for most object classes. MPEG-7 color descriptors generally perform best with original objects on black and white background (Regions 2 and 3). These regions are also the best choice for texture features when no accurate segmentation is given. At the first glance, white background outperforms black background on the given dataset but the results of the precision-at-k did not verify this assumption. Moreover, square bounding boxes always achieved better results than rectangle bounding boxes for SIFT and MPEG-7 features by an average increase of 2%. This indicates that the effect of changing the object's aspect ratio is worse than using a larger amount of background. However, for Gabor wavelets no significant changes have been measured between square bounding boxes and rectangle ones.

Segmentation accuracy: In order to simulate inaccurate segmentations from the given test set, we used the convex hull around perfectly segmented objects. Table 4.3 shows the classification results of perfectly and inaccurately segmented objects. These results indicate that accurate segmentation can improve the classification significantly (up to +24.5%) when the region is prepared appropriately. In contrast, only smaller improvements of about 2% are achieved between unmodified regions (Region 1) and modified ones for inaccurate segmentation. Only the results of Gabor wavelets improve from 20.5% to 25.3% and 24.8% for black and white backgrounds. Region 6 performs worse than all other regions for inaccurate segmentation because these regions only contain very rough object contours, as shown in Figure 4.2.

Feature types: The performance of SIFT and Gabor wavelets is similar for both segmentation accuracies and all regions except Region 1 and Region 4 where the background is left unmodified and blurred, respectively. Gabor wavelets perform slightly better on rectangular bounding boxes while SIFT achieves better results on square regions. MPEG-7 ColorLayout and ScalableColor features perform worse than texture features for the given task. Although Figure 4.5 indicates that ColorLayout outperforms ScalableColor this is only true because the best performing region

	Feature Type	***Region 1***	***Region 2***	***Region 3***	***Region 4***	***Region 5***	***Region 6***
Perfect Seg.	**SIFT**	25.0	38.3	40.4	32.0	29.8	**46.5**
	GaborWavlets	20.5	37.2	39.9	21.0	31.5	**45.0**
	ColorLayout	15.4	22.4	23.6	19.6	15.0	**28.7**
	ScalableColor	16.4	**21.8**	21.4	21.6	16.5	-
Inacc. Seg.	**SIFT**	25.0	27.2	**27.5**	22.5	27.2	12.1
	GaborWavlets	20.5	**25.3**	24.8	19.1	25.1	10.8
	ColorLayout	15.4	16.8	**18.6**	17.9	15.2	15.1
	ScalableColor	16.4	16.5	**16.8**	16.5	15.8	-

Table 4.3: Overall recall (in %) for perfect and inaccurate segmentation.

preparation approach (Region 6) is not applicable for pure color features (ScalableColor) where no spatial information is used.

Dissimilarity measures: The difference between the best and the worst dissimilarity measure for all features is about 3-5%. For instance, the results of SIFT features for Region 6 on perfect segmentations lie between 46.5% for the best (Jeffrey divergence) and 42.4% for the worst measure (Canberra metric). The highest variations are caused by MPEG-7 ScalableColor features. It seems that the ranking of dissimilarity measures does not depend on the used region preparation technique because the results of all measures are similarly ordered for all techniques. The best dissimilarity measure for all features was Jeffrey divergence followed by Chi-Squared statistics. The worst measure was Fractional distance for all features followed by Canberra metric for texture features. L1 metric performed best of the Minkowski family measures, especially for texture features where the difference to L2 distance was above 2.5%.

Discussion: The experiments have first shown that it does matter how the regions of segmented objects are prepared for semi-local feature extraction. Regions with modified objects and backgrounds can improve the overall classification rate significantly compared to unmodified regions, especially for accurately segmented objects. Secondly, square bounding boxes achieves better results than tight, rectangular bounding boxes. Thirdly, texture features perform better than color features and improvements of a few percent can be achieved when the right dissimilarity measures are chosen. The Jeffrey divergence and Chi-Square correlation performed best for all feature types and region preparation techniques.

4.1.5 Case Study

In addition to the evaluation based on perfect object segmentations, we performed a case study with the automatic image segmentation approach of [88]. This case study does not aim at the execution of an entire object detection workflow but it tries to discover potential challenges in the combination of non-perfect segmentation approaches with semi-local features. Therefore, we selected a segmentation approach that adds every image pixel to an image region without region classification. This approach leads to three different segment types. The first segment type only captures parts of an object or the entire object. The second type captures only non-object parts while both are captured by the third type, object parts and image parts that do not belong to this object. The third image of Figure 4.6 shows these segment types in gray, red, and yellow.

Experimental setup: The case study is done with images of the Pascal VOC test set including several instances of all 20 objects. In a first experiment, we used the default parameters of the automatic segmentation approach and extracted semi-local features for each segment. In a second experiment, we executed the segmentation several times for each image with different parameters. In this process, we got a couple of overlapping segments for each object similar to the multi-segmentation approaches that are explained in the related work section. First, we performed a manual inspection of the resulting semi-local feature regions, compare Figure 4.2. Then, we extracted the semi-local features of each image region for all proposed region preparation techniques and performed a nearest neighbor search against the same features of all residual

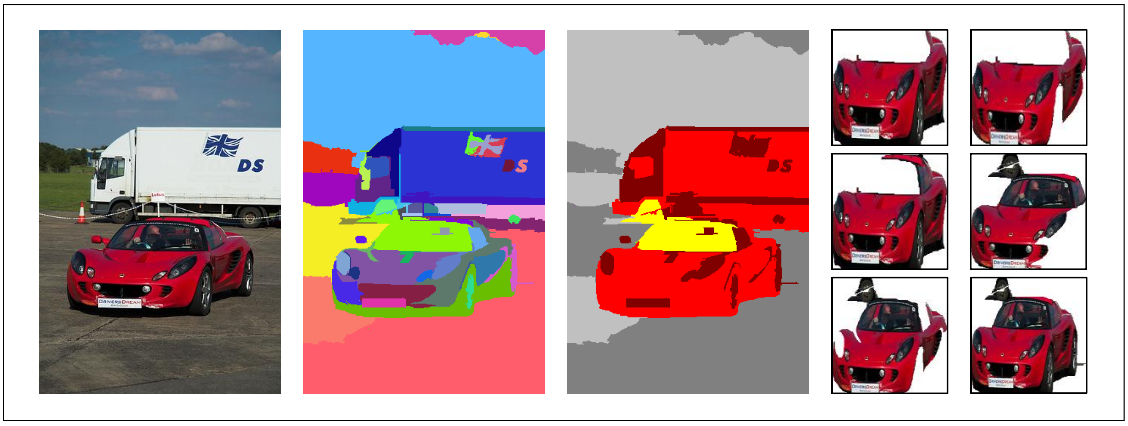


Figure 4.6: Semi-local features from image segments of [88]. (a) original image, (b) image segments, (c) segment types: gray contains only background, red contains only object parts, yellow contains both, (d) Region 3 examples from combined segments.

regions in the test set. In this process, we did another manual inspection of the matching regions. Note that no precision or recall values are given for this case study due to the small dataset size.

Observations: The image segments often capture the object boundaries accurately on some sites but seldom on all sites at the same time. Semi-local features that are extracted from these partially accurate regions do not often match with features of the same object. It only works if the rough object size and aspect ratio are preserved by the object segment whereby the results of Region 4 are the best ones. Sophisticated matching or region preparation techniques are required to improve this performance. Thus, we experimented with the combination of neighboring segments to one semi-local feature, as shown on the right side of Figure 4.6. Improved performance pays the price of increased complexity when more segments are combined to one semi-local feature and we counted up to a few hundred segment combinations per image. Furthermore, we observed that the segmentation robustness of specific object parts is good. For instance, the wheels of cars and buses were regularly segmented as individual regions. Semi-local texture features of Region 6 seems to be good candidates for object recognition with these object parts. The last observations consider the missing orientation invariance of the proposed semi-local features. If a large test set is given and the most common perspectives of each object are learned, orientation invariance is not important. Otherwise it is reasonable to rotate the image regions to their dominant orientation before feature extraction to gain rotation invariance similar to [191].

Discussion: From the object recognition view, two different challenges arise from these observations. The first challenge is it to figure out which segment (or combination of segments) captures a trained object and which ones only capture background. The second challenge is which segment captures an object best if many overlapping hypotheses are given. Obviously, this second challenge mainly arises if we roughly know where in an image we should search for the object. This kind of information might stem from other object recognition cues, such as a sliding window BoF approach. In order to tackle these challenges, we can either use perfectly segmented objects or automatically segmented objects to train the object recognition system. In the first case, only accurately segmented objects would result in correct matches. However,

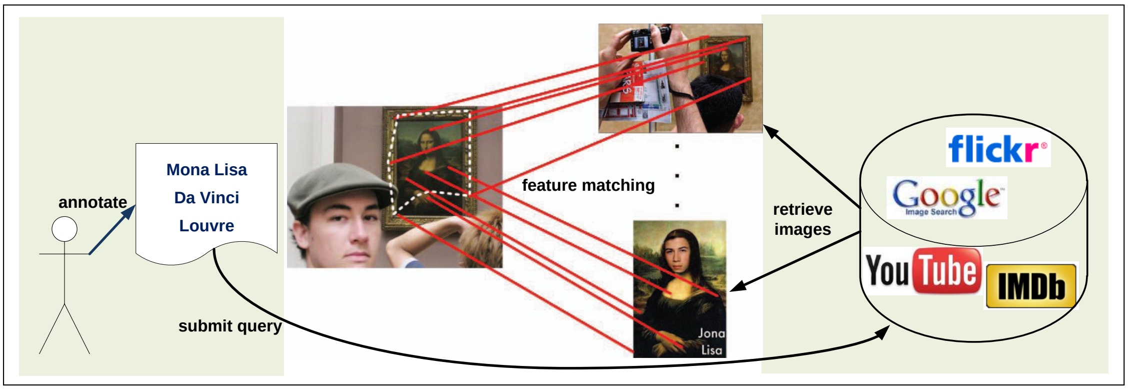


Figure 4.7: Object region detection is shown in the middle of the figure for a photo of the MonaLisa. The local feature matches with images of the same painting are used to form the region (white dashed lines). In the retrieval-based scenario (green background), users enter a textual annotation to the input image that is then used for image or video retrieval.

perfectly segmented training examples seem to be the appropriate choice to identify the best segmentation of overlapping hypotheses, like the ones in the right of Figure 4.6. In the second case, even partially accurate segmentations can lead to correct object recognition if they are segmented similarly in training and test images.

4.2 Object Regions from Local Feature Matches

In the last section, visual features are extracted from object segments and in the corresponding case study image segmentation was used as a pre-processing step. In this approach, entire objects are not often placed in one segment whereas object segmentation systems have been proposed recently that achieved accurate results for small sets of hand-selected objects (compare the PascalVOC segmentation challenge [85]). However, these state-of-the-art approaches are not applicable for segmentation of arbitrary objects, especially when only a few training examples exist. In this section, we propose a object segmentation strategy that uses a fairly different strategy. Its basic idea is that local feature matching can be used to select the object region of objects that are repeatedly shown in different images or video frames. In the first place, we do not consider how such images can be collected but cover this issue in Section 4.2.1 and in the video object annotation of Section 5.1. Generally, we assume that the proposed region detector needs to be resistant against a large number of images that do not contain the same object. However, two images of an object might be sufficient to generate a rough region while a larger number of such images increases the probability of accurate object regions.

In particular, the approach uses the following three steps. First, local features are *extracted* from all images. Feature *matches* between the currently investigated image and the residual ones are then computed before a *verification and clustering* takes place. In this process, an object region is generated from all feature matches that geometrically fit together. The middle of Figure 4.7 illustrates this procedure for images of the famous painting MonaLisa. The red lines show feature correspondences, the white dashed line represents the detected object region.

The object region detection can be applied with all local features described in Section 2.3 and the feature matching approaches of Section 2.4.1. After generating point correspondences between the investigated image and the residual ones, we apply geometrical constraints for verification. Each correspondence generates a hypothesis about the position of the center (x, y), size, and orientation of the matching image with respect to the investigated image. We compare the hypotheses of different point correspondences against each other and verified if they differ in x, y position and scale for less than 25%, and if the deviation of their orientations is less than 30°. These geometrical constraints are necessary because single correspondences can be accidentally generated while a lot of correspondences that fit geometrically together indicate a high probability that the same visual content is shown in both images.

Once at least 5 correspondences are geometrically verified a new object region is spanned over the convex hull of the matching interest points (or the centers of interest regions) in the investigated image. This geometric verification approach is similar to the generalized Hough transform proposed in [191] but it overcomes the problem that similar hypotheses can be situated in different bins due to fixed bin boundaries. In order to generate a joined region that is based on all the matching images, a point clustering is then iteratively performed. All correspondences are thereby projected into the investigated image, as shown for the left photo of the MonaLisa in Figure 4.7. A combined region spanned over the convex hull of all points is then computed if the hypothesis of the currently matched image overlaps with the existing hypothesis. Otherwise a new point cluster and region is generated for the hypothesis of the currently matched image. This approach is able to provide intermediate regions after each verification, and thus a termination criterion after a fixed number of successful verifications might be used for performance reasons. Moreover, this approach supports multiple appearances of an object in the investigated images, as shown in the clapperboard example in Figure 4.9.

4.2.1 Retrieval-based Region Detection

As mentioned before, the proposed approach is only reasonable if several images are given where the same object is shown. In the context of semi-automatic object annotation, object regions can be obtained after a user enters the name of an object or any textual annotation that she associates with this object. Feature matching is then performed with images or video frames that are retrieved from online platforms, such as Flickr, Google image search, and YouTube. This approach is shown in Figure 4.7 (green background) and is inspired from the AnnoSearch work of [334]. The detected regions have a clear visual relation to the input annotation, and thus lead to more specific, precise, and relevant annotations than global annotations. In this thesis, we developed a couple of image and video grappers to retrieve content using the open APIs of Flickr and YouTube, as well as an experimentally grapper to get images from IMDB that either belongs to a movie or an actor.

In order to test the proposed approach, we downloaded up to 250 images for 41 object labels from Flickr and try to apply it for each image individually. In the image retrieval, we used Flickr's *interestingness* sorting that provides a means for obtaining a good mix of images, as opposed to sorting by recent upload date or direct text matching. The 41 objects are chosen in accordance to [233] and they belong to the 6 categories fruits & flowers, monuments & buildings, brands & logos, famous paintings, and general objects. A complete list of all objects and the

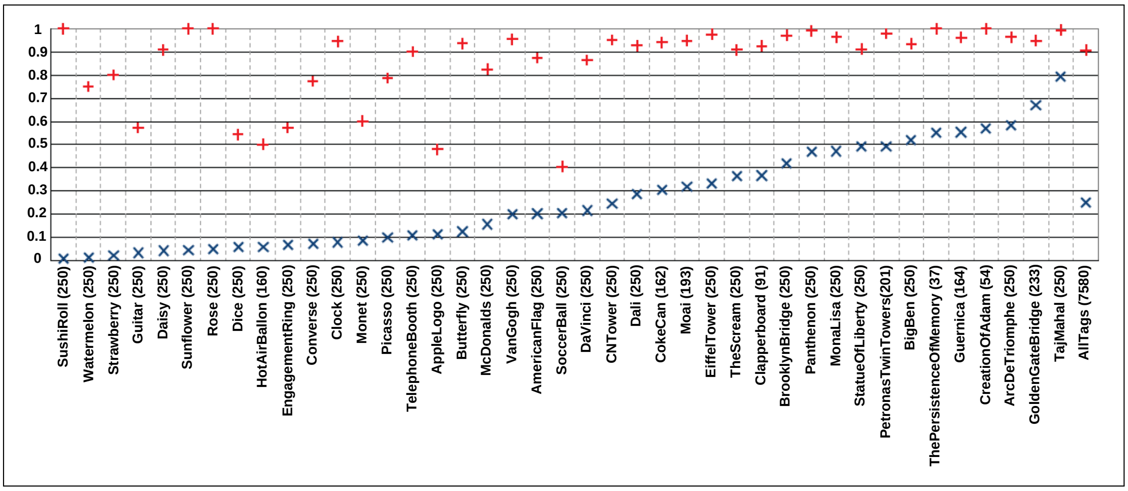


Figure 4.8: Precision (+) and recall (x) including object regions of the grades A1, A2 and A3 per object and for all objects (last entry).

number of downloaded images is shown in the bottom row of Figure 4.8. The actual number of images per object varies since for the more uncommon object labels only fewer images can be retrieved from Flickr. To get an idea of the difficult challenge that this dataset implicates, the reader is invited to browse the set of images on Flickr and convince himself of the extreme diversity of image content that are retrieved for an object label.

In this experiment, local feature matching is done with DoG points and SIFT descriptors [191], a high-contrast filter to select a maximum of 500 points, and a nearest neighbor distance ratio with Euclidean distance, see Section 2.4. We assessed the accuracy of detected regions by manually assigning them into one of the grades of Table 4.4, shown in the examples of Figure 4.9. After this, we computed the precision and recall of correctly detected regions for each of the 41 objects, and tried to determine the average accuracy of the regions. Thus, we matched all images of an object label against each other and classified them manually into one of the following categories: *True positive* if an object region was correctly detected according to specified accuracy grades. *False positive* if a region was detected although the object is not shown in the investigated image, and *false negative* if no region was detected although the object is shown.

Grade	***Accuracy***	***Explanation***
A1	very accurate	The region traces the boundary of the object-of-interest tightly.
A2	accurate	The region traces the object boundary with small over or under-size.
A3	moderate	The region hit the object but does not traces its boundary.
A4	poor	The region hits only parts of the object.
A5	false	The region does not encompass the object at all.

Table 4.4: The detected regions are assigned to one of these grades for evaluation.

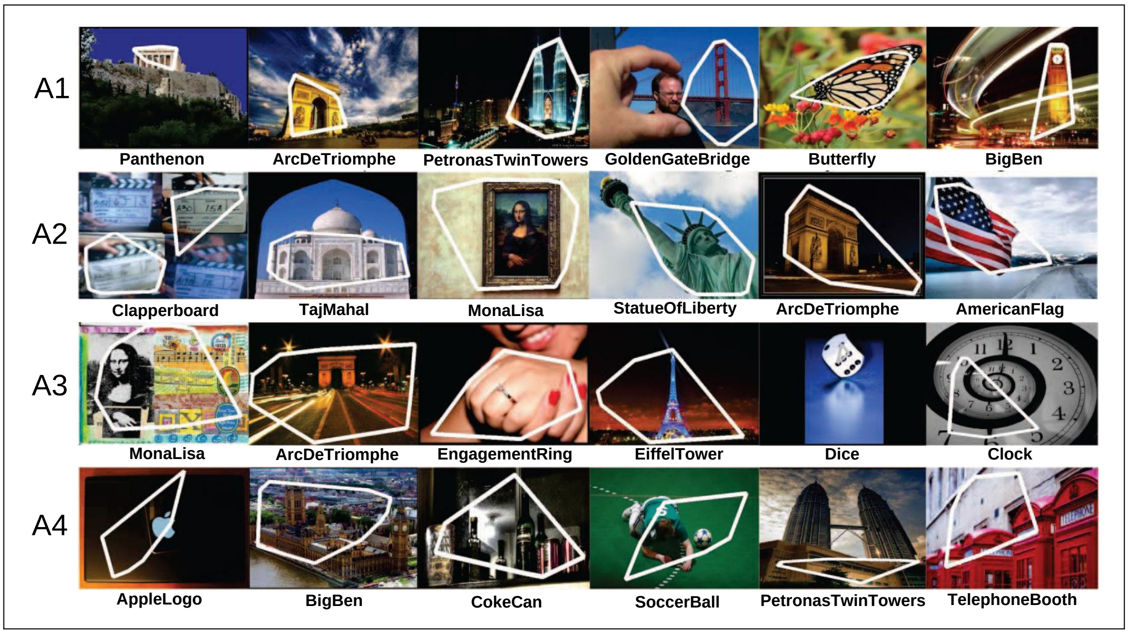


Figure 4.9: Exemplary results of detected regions for different grades (A1 to A4).

Examples of detected regions can be seen in Figure 4.9. The majority of the detected regions are of high accuracy (50% grade A1), followed by a large number of regions rated as A2 (31%), and less than 10% are clasified as poor (A4) or false (A5). All regions of grade A1, A2 and A3 are considered as true positives in Figure 4.8, where the precision and recall for all objects are shown. Generally, this experiment indicates a high average precision of 89%. The highest recall (80%) was reached for TajMahal, the averaged recall value is 25%. For the two categories monuments & buildings and paintings an average recall of almost 50% was reached. On the contrary, the approach shows difficulties with the categories fruits & flowers and general objects as their images have a high visual variability whereas images of famous paintings or buildings are often taken from similar viewpoints for capturing the object-of-interest in a prominent position. Moreover, false positive detections hardly influence the accuracy, with only 2% false positives in average.

4.3 Motion Features

Two different motion detection approaches are further investigated in this thesis that are especially designed for action scene detection of Section 6.3. The general idea of both features is to capture motion using the difference between a video frame and frames of the following second. This difference is small if no or little motion exists between the investigated frames and it is high if high motion or shot boundaries exist. In this way, a single motion feature is generated for each video frame. The length of one second was chosen to avoid that multiple shot boundaries are situated within an interval. In consequence, the shot structure of a movie is implicitly captured

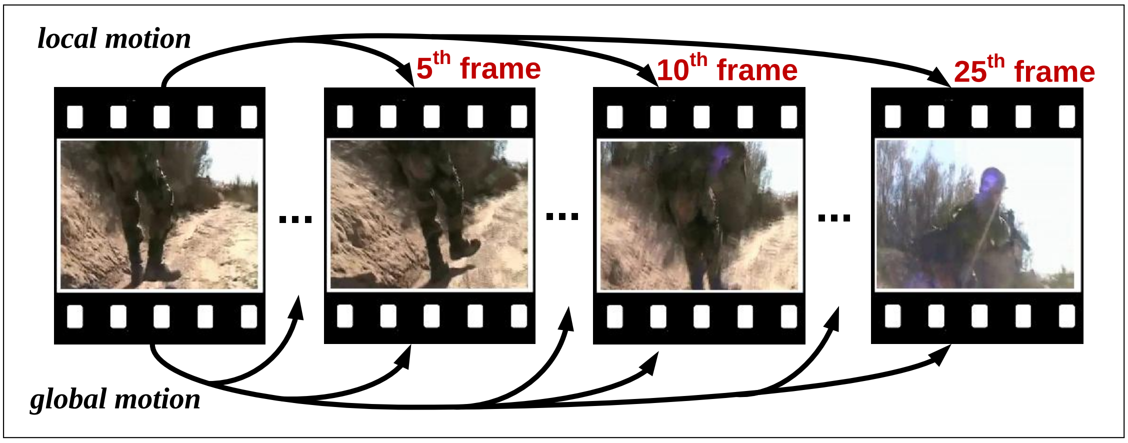


Figure 4.10: Motion features extraction. The difference of an investigated frame (leftmost one) to the frames of the following second are captured by global gist (lower arrows) and local SIFT features (upper arrows).

without shot boundary detection and only a few frames are affected by shot boundaries in scenes with little motion.

4.3.1 Global Motion

First, a global gist feature [231] is extracted from every frame of the video. This feature gathers the image gradients of sub-regions in 16 orientation histograms with 8 bins each. A simple motion feature is then generated for every video frame by comparison with the features of the 25 successive frames using Euclidian distance. The arrows in the bottom part of Figure 4.10 show this task for the leftmost frame in the film strip. The resulting motion features have 25 dimensions.

4.3.2 Local Motion

While the global motion is suitable to gather the general amount of change in a scene, local motion can be used to describe what is going on in a scene on a much finer level. It collects information about the motion of different objects in relation to the camera movement. In contrast to global motion, we extract local motion characteristics of one second by comparison of the starting frame to the 5th, 10th, and 25th following frames instead of using every frame. This is shown in the upper part of Figure 4.10. Since such a sparse frame matching would hardly work with classical tracking or optical flow approaches, local SIFT descriptors are used instead. In preliminary experiments, the given frame intervals have performed best.

We extract SIFT descriptors from Difference of Gaussian (DoG) points [191] in every frame to compute local motion. A filtering step is used to limit the maximum number of DoG points per frame to 350. Thereby, points with low contrast to their scale-space neighborhood are rejected because they tend to be unstable over time compared to high contrast points. Frames with less than 10 descriptors (e.g. black frames) are totally rejected. Similar to the original SIFT approach, a SIFT descriptor of the investigated frame is re-detected in a following frame when the relative

Euclidian distance between the nearest neighbor and the second nearest neighbor is higher than 60%. Finally, the number of re-detections between a frame and its 5th, 10th, and 25th successor are used as first 3 dimensions of the resulting motion vector. This vector further includes the ratio between the first dimension and the other ones (the number of re-detections of the 10th and 25th frame compared to the 5th frame). In preliminary experiments we also investigated more complex statistics to describe local motion considering the geometric correspondences of matches and the actual motion change (translation, scale, orientation), but the results indicated that the simple statistics perform better.

4.4 Summary

This chapter presents a couple of new visual features and feature extraction strategies. First, semi-local features are proposed for object recognition using segmentation as a pre-processing step. In this approach, state-of-the-art texture and color features are extracted from regions that cover the entire object with and without background modifications. Results of an extensive evaluation indicate that semi-local features are good candidates to improve object recognition systems. The experiments investigated perfect segmentations and inaccurate ones, on the one hand, and automatically segmented image regions, on the other hand. The classification was done with a nearest neighbor matching strategy and different dissimilarity measures to keep the evaluations as simple and universally valid as possible.

In this process, we have first shown that it does matter how the regions of segmented objects are prepared for semi-local feature extraction. Regions with modified objects and backgrounds can improve the overall classification rate significantly compared to unmodified regions, especially for accurately segmented objects. Moreover, square bounding boxes always achieves better results than tight rectangular bounding boxes, and texture features perform better than color features. Improvements of a few percent can further be achieved when the right dissimilarity measures are chosen. These findings contribute to object recognition research as only the work of [305] contains further segmentation-specific features.

After this, we proposed an object region detector for arbitrary objects that are repeatedly shown in images and video frames. The obtained image regions are intended to extract semi-local features or as alternative cue for content-based object queries in video retrieval systems. The detector is based on local feature matching, geometric verification, and furthermore a retrieval-based variant is given that builds on online image and video portals. The performance evaluation on 41 different objects and about 10000 images shows that the proposed technique produces results with a high precision of 89% and a successful region detection in one image out of four. The detection is object-independent and it works fully unsupervised, without time-consuming offline feature extraction or training. The work which is most closely related is [244] that projects image regions from one image to another using homographies found during local feature matching. In contrast to the proposed detector, they try to join different views of the same object.

Finally, we present two motion features that capture statistics of one second video with global gist features and local SIFT features, respectively. These features are very compact and enable action scene detection without shot-boundary detection, as described in Section 6.3.

However, they might be well suited for a broader range of video annotation and retrieval tasks, especially the global variant that is extremely fast to compute.

CHAPTER 5

Object-of-Interest Annotation

In this chapter, we address the use of object recognition techniques to annotate *what* is shown *where* in video collections. These annotations are suitable to retrieve specific video scenes for object related text queries which is not possible with the manually generated metadata that presents the current standard. We are not the first to present object annotations that are generated with content-based analysis methods, see Section 2.1. However, the proposed approach possesses some outstanding features that offer good prospects for its application in real video annotation and retrieval systems. Firstly, it can be easily used as background module in any video environment. Secondly, it is not based on a fixed analysis chain but on the extensive recognition infrastructure of Chapter 3 that can be used with all kind of visual features, matching and machine learning techniques. New recognition approaches can be integrated into this infrastructure with low development costs and a configuration of the used recognition approaches can be performed even on a running system. Thus, it might also benefit from future advances in computer vision. Thirdly, we present an automatic selection approach to support the use of different recognition strategies for different objects.

A freely available prototype is then presented and a case study is performed with professional and user-generated videos. This evaluation focuses on the object-of-interest annotation in real-world scenarios and on the automatic approach selection. Finally, we describe how this approach might be integrated into existing video platforms.

5.1 Approach

This section presents an easy and user-friendly approach to annotate those scenes in a large video collection that contain a specified object-of-interest. These annotations are well suited to improve text-based video retrieval, especially when a user searches for specific objects or scenes. In this process, we exploit object recognition approaches without presenting new methods or fixed analysis workflows. Recognition approaches of all kinds including global and local detector-descriptor chains, matching strategies, and machine learning techniques can be inte-

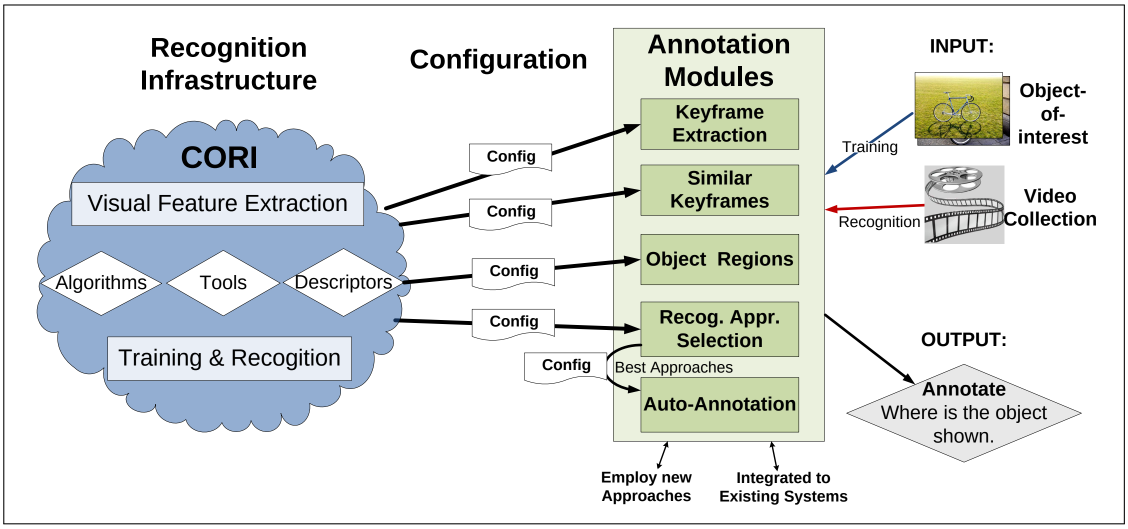


Figure 5.1: The recognition infrastructure CORI is used for all annotation modules and their actually used visual features and recognition approaches are adjustable with a configuration. On the right figure side, the input and output of the object-of-interest annotation are given.

grated and the annotation approach can be easily extended to incorporate computer vision advances. This flexibility is achieved by the recognition infrastructure CORI, see Chapter 3. As shown in Figure 5.1, all annotation modules are based on CORI and the actually used features and algorithms can be adjusted by a configuration. Despite its flexibility in matter of recognition approach extension and its compatibility with existing annotation and retrieval systems, a few cornerstones mark the approach's efficient nature: Firstly, videos are represented by a few keyframes per scene. Secondly, object annotations and low-level features are stored in a simple and compact form with sufficient information to answer text queries on scene level. Thirdly, an appropriate recognition approach is automatically selected for each annotated object. Moreover, the relevance of scene and object level retrieval results should then be visible to the user at the first glance.

The left part of Figure 5.2 shows the proposed annotation workflow including the required user interaction and the generated data. In this workflow, the entire video collection is first *preprocessed* to extract keyframes and visual features. User interaction is then required to initiate the annotation of new objects by an interactive collection of a few object instances starting from a single keyframe that shows this object. In this process, keyframes that are supposed to contain the same object are returned to the user for feedback and the region of each object instance is automatically selected. After this *object specification*, appropriate *recognition approaches are selected* that fit to the visual attributes of the object in an integrated analysis-evaluation approach. Eventually, the object is *automatically annotated* in all videos of the video collection.

The right part of Figure 5.2 shows the database schema that is used to store the results of this object annotation. *Objects-of-interest* are given with one or more user-generated free-text labels to enable retrieval thesis. These labels can be ambiguous as the same object might be annotated differently [201], but it is out of the scope of this work to dissolve such ambiguities. Furthermore, metadata (like a user identification, annotation date, and an interaction protocol) are stored

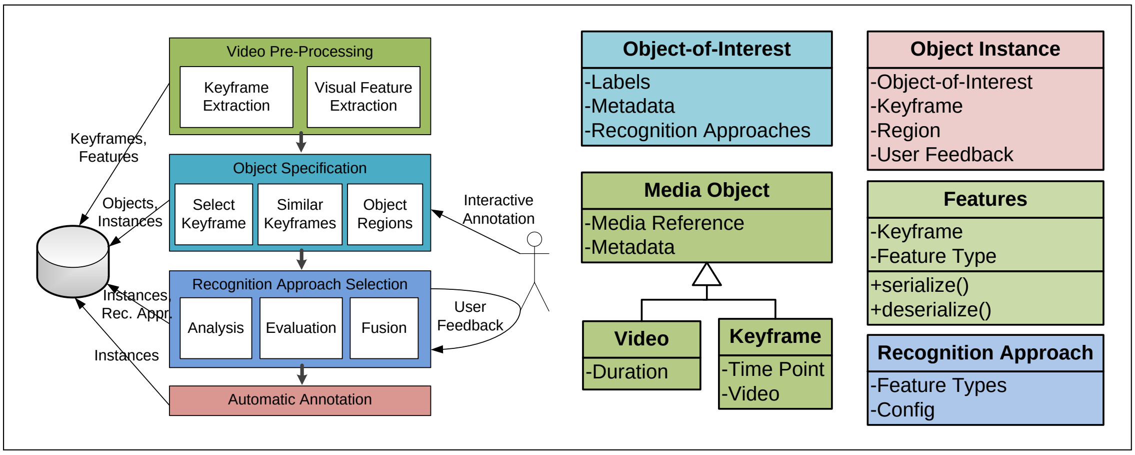


Figure 5.2: The annotation workflow is shown on the left with all processing steps, user interaction, and data storage. Thereby, instances of the associated database schema (right) are used.

together with the selected *recognition approaches* that are used for the automatic annotation of this object. *Object instances* capture the object-of-interest that is shown in a keyframe with an optional region (arbitrary shaped polygon) and a user feedback flag (correct or false annotation). *Videos* and *keyframes* are derived from the *media object* class and they contain a media reference and metadata. The keyframes are either stored in small resolution or extracted on the fly from the video at the specified time point. In the latter case, the media reference is left empty. On the fly generation is especially reasonable when this annotation approach is integrated to a system that contains suitable video summarization and browsing capabilities. The schema can be extended by further media object entities, such as shots and scenes, if necessary. Shot entities are, for example, used in the TRECVID experiments of Section 6.1. Furthermore, the proposed annotation schema contains *visual features* that are serializable to binary form which makes them suitable for storage in all kind of databases and file systems. In the following, all annotation steps of Figure 5.2 are explained in detail.

5.1.1 Video Pre-Processing

As first step, all videos are pre-processed in an off-line process to enable an efficient annotation process. Keyframes and visual features are thereby extracted from the videos without any user interaction. We expect that a large number of videos has to be processed in this process and that new videos are frequently added. Thus, pre-processing can be performed in a distributed manner on multi-core and multi-machine architectures. The generated data is stored in the database schema of Figure 5.2.

Keyframe extraction: Keyframes are a very common way to present videos and they are often used as input for content-based video analysis. The keyframe extraction approach of [316] and others is similar to the one of this thesis. A standard shot detection [182] is applied before an activity-based keyframe selection algorithm is performed for each shot. Shots with little activity are summarized with only one or a little keyframes while up to 10 keyframes are extracted for high-activity shots. We measure the activity with global SIFT descriptors similar to the work of

Algorithm 5.1: Keyframe Selection

```
Input:
    vector<keyframe> $frames$ = every $10^{th}$ frame of the shot
    float $threshold = $ threshold of last shot
Output:
    vector<keyframe> $keyframes$

    repeat
        $keyframes$.clear()
        for all $f$ in $frames$ do
            boolean $lowActivity$ = false
            for all $k$ in $keyframes$ do
                if EuclidianDistance( globalSIFT($f$), globalSIFT($k$) ) $< threshold$ then
                    $lowActivity$ = true
                end if
            end for
            if $lowActivity$ == false then
                $keyframes$.add($f$)
            end if
        end for
        if $keyframes$.size() $<= 2$ then
            $threshold = thershold$ * 1.25
        else
            $threshold = thershold$ * 0.95
        end if
    until $keyframes$.size() $<= 10$
```

Oliva and Torralba [231] that used global features to capture the gist of a scene. As shown in Algorithm 5.1, the Euclidian distance is used to compare these features and an adaptive threshold that is decreased until no more than 10 keyframes are selected for a shot. During analysis of the first few video shots an appropriate value is automatically selected for this threshold, and thus its start value (default 0.5) only affects these first shorts. Due to the simplicity of these global features, keyframe selection performs faster than real-time.

Feature extraction: The selected keyframes are then used to extract various visual features for object specification, recognition approach selection, and object annotation. Note that the proposed system is not restricted to the use of specific features and recognition approaches, but the following features are currently integrated to the system: SIFT [191], SURF [211], Gabor Wavelets [99], MPEG-7 DominantColor, ColorLayout, ColorStructure, and EdgeHistogram [199] features. These features can be extracted globally from the entire keyframe or locally from dense sampled regions [146], Difference of Gaussian points [191], MSER regions [310], Viola–Jones faces [327], histogram of oriented gradients (HOG) body regions [59], and from image segments of different segmentation techniques, see Section 4.1.3. Furthermore, a set of image pre-processing, region filtering approaches, and object models are used. As feature extraction takes time and feature storage takes memory, it is reasonable to select only a subset of these features for analysis of specific video domains. CORI enables the selection of such subsets even on a running system without high development and redeployment efforts.

Algorithm 5.2: Suggest similar keyframes

```
Input:
    vector<features> $positives$ = features of the selected keyframe
    vector<features> $negatives$ = features from keyframes without the object
    integer $videoId$ = video id of the selected keyframe
    integer $keyframeId$ = id of the selected keyframe
Output:
    vector<keyframe> $suggested$

    for $i = 1$ to numberOfKeyframes( $videoId$ ) do
        $features$ = loadFeatures( $keyframeId + i$ )
        $distPos$ = nearestNeighbour( $features, positive$ )
        $distNeg$ = nearestNeighbour( $features, negative$ )
        if $distPos > thresholdA$ and $distPos/distNeg < thresholdB$ then
            $suggested$.add( $keyframeId + i$ )
        end if
        $i = -i$
        if $i < 0$ then
            $i = i - 1$
        end if
    end for
```

5.1.2 Object Specification

In the second annotation step, users can interactively select a few keyframes that contain the object-of-interest. These keyframes are used to select an appropriate recognition approach including region detector-descriptor chains, matching strategies, and dissimilarity measures, as well as the parameter settings of all components. Thus, it is important to collect examples of the object that represent the appearance of this object in all videos of the collection well with appropriate difficulties and levels of abstraction. If we want to annotate all cars independent of the used camera view, for example, then the initial object-set should not only include red sport cars shown from a frontal view. The object specification is composed of following three steps.

Select keyframe: First, a single keyframe of the object-of-interest has to be selected in one of the videos. This can be done when the user is browsing through the extracted keyframes, while she is watching a video, or in any other way that is provided by the surrounding video system. The prototype that is presented in Section 5.2 allows keyframe selection in a simple interface where the thumbnails of an entire video are shown. The actual position of the object within a selected keyframe is automatically computed in a later processing step, described below. Generally, examples of an object might also be used from outside the video collection, but this is not supported yet because we need to know from which video and keyframe the first object example stems.

Similar keyframes: After the user has selected a keyframe that shows the object, we start to search for further examples of this object from nearby keyframes according to Algorithm 5.2. This search starts in the video of the selected keyframe but, if necessary, it can be expanded to related videos that stem from the same user or that have been labeled with similar tags. Due to the high run-time requirements of this process (the user is actively waiting for results to mark them as positive or negative examples) we use a fast bag-of-features matching (BoF) with dense sampled SIFT and ColorLayout features. Feature matching of an investigated keyframe

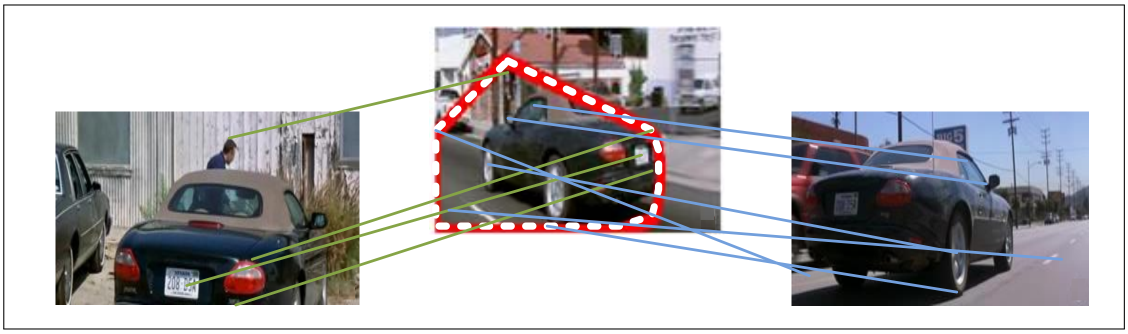


Figure 5.3: Automatic detection of object regions from a few keyframes. The region detector of Section 4.2 is applied in this process to match nearby video frames before frames of different videos are investigated.

K against all positive and all negative object instances O is thereby performed with a nearest neighbor strategy and the histogram intersection:

$$histogramIntersection = \sum_{i=0}^{i<cbSize} \max\left(K_i, O_i\right) - O_i \tag{5.1}$$

where i is the current codebook index of both BoFs. This histogram intersection decrease the importance of background objects that are shown in both images compared to traditional distance measures such as Minkowski distances and statistical measures [189]. Keyframes are suggested to the user if the distance of the current BoF to its nearest positive neighbor is below a certain threshold and the ratio between the nearest positive neighbor and the nearest negative neighbor is above another threshold. The suggested keyframes are shown to the user as soon as they are available. Instantly after the user provides feedback (if a keyframe contains the object-of-interest or not), the visual features of this frame are also used for further keyframe suggestions. This process is terminated after a few object examples have been identified. Due to the needed user interaction, we designed the Algorithm 5.2 in a way that a small number of object examples (typically between 2 and 6) is sufficient.

Object regions: After keyframe selection, we try to compute the rough position of the object-of-interest in these keyframes automatically with a relaxed region matching. In this process, we use SIFT descriptors [191] extracted from up to 300 Difference of Gaussian (DoG) points using a high-contrast filter in order to match each keyframe of the set against each other. This matching is done with a nearest neighbor distance ratio strategy using Euclidian distance as proposed in the original SIFT approach but with lower threshold ratios. Thus, it is also possible to generate matches for less similar descriptors that stem from objects of the same class. After this, we compute the object region in all selected keyframes with the convex hull around all matching DoG points, according to the object region detector of Section 4.2. The result of this region matching are shown for the top keyframe in Figure 5.3.

Algorithm 5.3: Recognition Approach Selection

```
Input:
    vector<recognition approach> $approaches$ = recognition approaches to investigate
    vector<instance> $objects$ = the initially selected object instances
    vector<keyframe> $collection$ = videos to annotate
Output:
    vector<instance> $results$

    for all $a$ in $approaches$ do
        vector<features> $resultsA$
        for all $leftOut$ in $objects$ do
            if $leftOut$.positiveExample() is false then
                continue
            end if
            $query$ = $objects$ - $leftOut$
            $testset$ = $collection$ + $leftOut$
            $orderedResults$ = analysis( $query$, $dataset$ )
            $position$ = evaluate( $orderedResults$, $leftOut$ )
            $eval$.add( $position$ )
            $resultsA$.add( $orderedResults$ )
        end for
        $result$.add( fusion( $resultsA$ ) )
    end for
    vector<setup> $bestApproaches$ = selectBest( $eval$ )
    $results$ = fusion( $results$, $bestApproaches$ )
```

5.1.3 Recognition Approach Selection

In the automatic approach selection, we compute the results for different features and recognition approaches, evaluate them, and combine the best ones. Optionally, we further present the results to the user to get a feedback. As shown in Figure 5.2, this process consists of the three steps: *analysis*, *evaluation*, and *fusion*. The initial object examples are used to compare different recognition approaches against each other and, if user feedback is given, these steps are iteratively executed to improve the achieved results.

As specified in Algorithm 5.3, appropriate approaches are selected after the recognition results of all approaches have been individually generated and evaluated. This algorithm uses the popular leave-one-out strategy [132] to match all object examples except one against the keyframes of all videos and the left-out object in each iteration, see Section 2.4.2. The position of this left-out object within all results is then used for evaluation. The algorithm can be applied to analyze a few hundred hours of video at once. However, it is also reasonable to start the analysis on a single video. In each iteration, an ordered list with object instances is returned and one combined list per recognition approach is then generated using the same fusion that is later used to combine the selected recognition approaches for the investigated object. In the following, details about analysis, evaluation, and fusion are given.

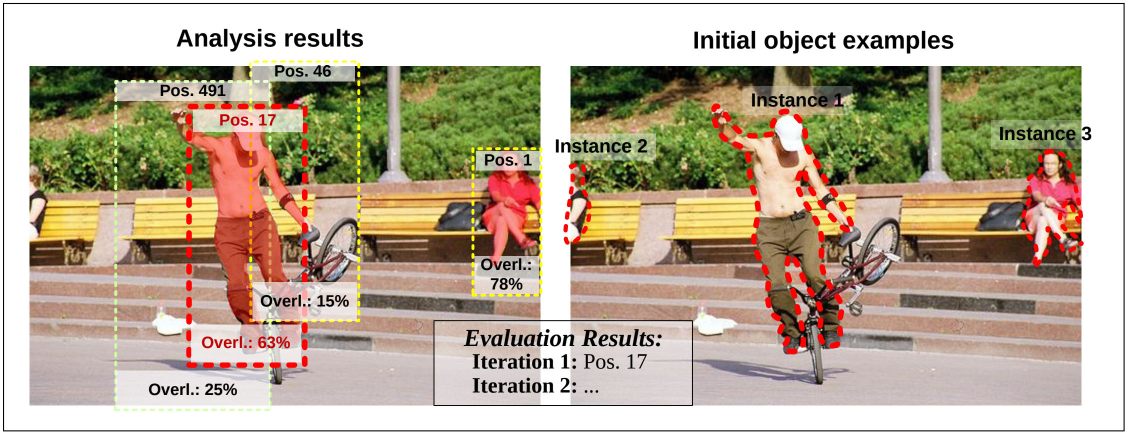
Analysis: In the analysis step, we try to recognize the selected object in all keyframes with a couple of different recognition approaches. Object instances are generated for each recognition approach and ordered according to their recognition probability. Instances with a high recognition probability are given on top of this list. As mentioned before, analysis is not restricted to specific recognition approaches, but the following approaches have been used in the experiments of this thesis with the features mentioned in Section 5.1.1. On the one hand, we have the matching strategies: nearest neighbor, k-nearest neighbor, nearest neighbor distance ratio [191], thresholding as well as their combinations. All of these strategies have been used with the dissimilarity measures Manhattan distance (L1), Euclidian distance (L2), Canberra metric, Jeffrey divergence, Psi Square, cosine function based similarity, and histogram intersection. On the other hand, we used SVMs [132] with different kernels and applied a few post-processing steps, like geometric verification and feature voting. Each approach further contains a number of parameters, such as the threshold of simple feature matching, that can be tuned to improve the recognition of specific objects. Further details about the actual recognition approaches are given Chapter 2.

Evaluation: If no region information is given in the initial object examples, a positive match is generated when the left-out keyframe is contained in the analysis results and we return the position of this match. This keyframe level comparison is possible because every object instance includes the information in which keyframe it was detected, as shown in the annotation schema of Figure 5.2. When region information is given, all object instances are individually compared against the left-out object by computation of the region overlap using a polygon intersection with the evaluation framework of Section 3.4. Figure 5.4 shows this evaluation for a task where three object instances ('persons') have been selected (right image) with exact object boundaries. The suggested object instances from a recognition approach (left image) are given as bounding boxes. In this figure, the person shown in the middle of the keyframe was used as left-out object. Thus, the evaluation tries to find out if this object instance was returned and at which position in the ordered result list. During evaluation, every object instance that was generated from this keyframe is compared against the left-out instance using polygon intersection. Those instances that are above a certain intersection (default value is 60%) are true positives (bold bounding boxes) and all other are false positives (thin bounding boxes). The position of the best true positive is then returned, which is 17 in the shown example. Note that the proposed evaluation process does not guarantee that the best recognition approaches are selected. However, the case study of this chapter shows that it works well if the initial object examples present the object in the videos well.

Fusion: Finally, the one-left-out results of each approach as well as the recognition approaches that achieved the best evaluation results are combined using following fusion strategy.

$$rank_i = best_i * \#lists * \frac{\#entries_i}{\#possibleEntries} - best_i \tag{5.2}$$

where the rank of each object instance *i* in the result *lists* is based on its *best* position in all result lists and the *number of entries* of this instance in all lists in relation to the *possible amount of entries*. Depending on the recognition approach, multiple entries of the same instance can either be given in one result list or not. After this, we order all instances according to these ranking values. The intuition behind this fusion is to order the instances similar to a zip fastener starting from their best entries. The order of entries with the same best value (originating from different result lists) is determined from the total number of entries. In an alternative fusion approach, we additionally used the number of top *x* ranked entries (x $\in$ 50, 100, 500).



Evaluation Results:
Iteration 1: Pos. 17
Iteration 2: ...

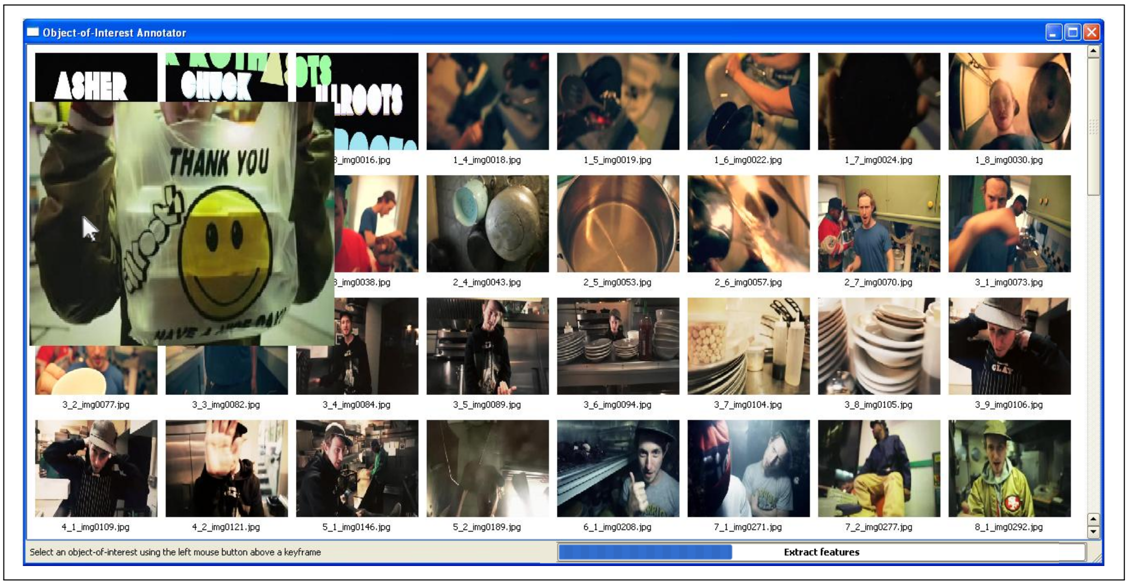
Figure 5.4: Evaluation for the first iteration of the one-left-out approach. Compare the evaluation results of this figure with the ones of Figure 3.10 to see the selection approach extensions for ranked result lists.

5.1.4 Automatic Annotation

The proposed recognition approach selection is only efficiently applicable for small video collections because the analysis of each approach takes some time, compare Section 3.4.4. Thus, we annotate only a couple of videos in this process and propose the annotation of further videos in an automated post-processing step. During this analysis, the selected annotation process is used to annotate an object only in the best matching videos. In contrast to the earlier steps of the workflow, we suppose that this annotation is performed without a human in the loop and that results of this annotation might only implicitly be inspected during video retrieval. The annotation of false object instances in this step will not significantly decrease the video retrieval performance, as user-generated annotations are always preferred to answer video queries. The automatic annotation process can be easily distributed over an arbitrary number of analysis machines, which is especially important for large-scale video collections that are incrementally growing.

5.2 Annotation Prototype

We developed a prototype that integrates the proposed annotation approach and performed a case study with two video datasets on it. This prototype can be downloaded from [325] and the reader is encouraged to perform experiments with his own videos. The video annotation prototype is implemented as client application with a simple user interface. It is given as Windows distribution and was developed in C++ with the Qt application framework [253]. The user interface of the prototype appears very sparsely to suggest the *touch and feel* impression of a web application that is shown in a usual browser. The only client application item in the prototype is a status bar that displays interaction hints and that contains a progress bar to visualize the current processing state. After a video was selected by the user for the first time, shot boundary detection and keyframe extraction is performed. In this process, the free Shotdetect software [279] and FFmpeg [91] are executed in parallel before keyframe clustering takes place. Next, the pro-



Select an object-of-interest using the left mouse button above a keyframe

Figure 5.5: Annotation prototype: Keyframes are shown in a compact keyframe view and the one under the mouse cursor is enlarged. This keyframe is selected as positive or negative examples by a left or right mouse button click and appears consequently in the top row of the keyframe view. The status bar displays interaction hints and a progress bar to visualize the current processing state.

totype extracts visual features from these keyframes and they are stored in a sub-directory of the video file. When a video is opened for the second time, the clustered keyframes and extracted features are loaded without further analysis.

As shown in Figure 5.5, the clustered keyframes are shown in a compact matrix form in a scroll widget with 8 thumbnails per line. Typically, 5 to 7 lines fit to the screen which means that a user can see about 50 thumbnails at once. This representation makes it easy to navigate over a large number of keyframes. Below each keyframe a label is displayed that consists of the cluster id and a reference to the video timepoint. A detailed view of a keyframe appears when the mouse cursor is placed above its thumbnail and users can start to annotate objects by pressing the left mouse button above a thumbnail. The selected keyframe is then shown in the first line of the widget and the suggested keyframes that are supposed to contain the same object are shown in the lines below. Users can select further keyframes as positive or negative examples using the left and right mouse button above a keyframe. Negative examples are also shown in the first line of the widget but starting from the right side. As described in Algorithm 5.2, keyframe suggestion uses both positive and negative examples, and thus we recommend the annotation of negative examples especially in those cases where the object-of-interest is not contained in the first suggested keyframes. Moreover, negative examples are helpful if two or more objects are shown on several positive examples although only one of them is the object-of-interest. This happens frequently during the annotation of a person because different persons are often shown together and in front of the same location. As soon as two positive keyframes have been selected, the region detection algorithm of Section 5.1.2 takes place in the background

```
[Visual Features]                           | [Recognition]
name     = Gist                             | name     = KeyframeExtraction
output   = FloatVector                      | input    = Gist
module   = imagePreprocessing, size_32_32;  | strategy = Nearestneighbor
           SIFT                             | distance = Euclidian
------------------------------------------- | ---------------------------------
[Visual Features]                           | [Recognition]
name     = BagOfFeatures                    | name     = SimilarKeyframes
output   = IntVector                        | input    = BagOfFeatures
modules  = denseSampling, s_0.5_0.3_0.15;   | strategy = K-Nearestneighbor
           SIFT; ColorLayout; BoFGenerator  | distance = HistogramIntersection
------------------------------------------- |  --------------------------------
[Visual Features]                           | [Recognition]
name     = localSIFT                        | name           = ObjectRegions
output   = FloatVector, Regions             | input          = localSIFT
modules  = denseSampling; DoG;              | descriptors    = FloatVector
           SIFT, highContrastFilter_300     | strategy       = NNDR, ratio_0.3
                                            | distance       = Euclidian
                                            | postprocessing = ConvexHull
------------------------------------------- | ---------------------------------
[Visual Features]                           | [Recognition]
name     = Persons                          | name     = ApproachSelection
output   = FloatVector                      | input    = {BoF,Gist,Persons}
modules  = FaceDetection; HoG;              | strategy = SVM, kernel_{RBF,X2}
           GaborWavelets                    | gamma_{1:+0.1:5}, c_{1:*2:8}
```

Configuration 5.1: Used setup of visual feature extraction (left) and recognition (right) for the annotation modules of the prototype.

to identify the object regions in all positive keyframes. These regions are then shown as red polygons in the prototype.

Users can finish the annotation of an object by pressing any button on the keyboard. After a label has been entered for this object, users can annotate further objects starting from the original keyframe representation. For each annotation, the prototype stores a few information (like the object name, the selected keyframes, and the needed annotation time) in simple text file. Moreover, it is possible to configure the used visual features and recognition approaches for the algorithmic modules keyframe extraction, similar keyframes, object regions, and approach selection of the prototype according to Configuration 5.1. Global SIFT features (*keyframe extraction*), a bag-of-features approach with dense sampled SIFT and ColorLayout features (*similar keyframes*), and local SIFT features from Difference of Gaussian (*DoG*) points (*object regions*) are used for these tasks. However, new algorithms and recognition approaches can be integrated into the proposed recognition infrastructure and they can be used without recompilation of the prototype simply by replacing one dynamic linked library (AnnotationFW.dll).

5.3 Case Study

We evaluated the interactive annotation process and the proposed algorithms of Section 5.1 in a case study with the prototype using two video datasets. The first set consists of 20 short videos that have been randomly downloaded from YouTube. The second set contains 10 longer videos including feature films, documentations, and an animated video, as listed on top of Figure 5.6.

Although the overall runtime of these videos is only about 15 h, we captured many characteristics of the annotation process including the number of average annotations per video, the needed user interaction time, and the storage requirements. We assume that these characteristics also apply to large video collections because each video is independently annotated while the automatic annotation of multiple videos (Section 5.1.4) is not considered in this case study.

5.3.1 Interactive Annotation

Table 5.1 gives an overview of the used video datasets. As shown, the average video duration of the short videos from YouTube was about 5 min with the shortest video of 31 s and the longest of 22 min. An average number of 68.5 keyframes are clustered from these videos, which means that on average one keyframe is used every 4.4 s while a larger interval of 7.6 s is given in the second dataset. The different cluster rates of the two datasets indicate a higher change frequency of the short videos. In each video, we tried to annotate a few objects that seem to be important for the entire video. Overall, we annotated 153 different objects in both datasets combined. The number of annotated objects per video highly depends on the video duration, and thus almost 10 objects are annotated for an average video of the second dataset while less than 3 objects are annotated in the short videos. In both datasets persons are the most frequently annotated objects followed by the category *location* that includes soccer fields, beaches, and specific buildings. The residual annotations either depict specific objects or object classes. Frequently annotated examples of these categories are specific cars (e.g. the car of a main actor) and a more general class of cars, such as taxis. The bottom of Figure 5.6 lists the labels of 88 annotated objects from the second dataset.

The variation of the used annotation time between the two datasets is rather small (33 to 47 s per object annotation) considering the large difference in size of clustered keyframes. The given annotation times include the selection of all positive and negative examples, and thus the time that a user needs to identify a new object that she wants to annotate. Obviously, it takes a longer time to select the first positive example when a larger number of keyframes are shown to the user and when more objects are annotated in one video. Considering the higher number of positive and negative examples that are selected for the second dataset, we conclude that the proposed interactive object annotation works efficiently even for entire feature films. Longer annotation times (up to 146 s) are often caused by the fact that no keyframe was suggested in the top ranks by Algorithm 5.2 that contains the initially selected object-of-interest. Furthermore, the average annotation times are slightly longer (55 s) for the gray-level movie *Alexander Newski* because object identification is more difficult when no color information is given.

Dataset	**#**	**Duration (min)**			**Clustered Kfs**			**# of Annotations**				**Categories**				**Anno. Time (s)**			**Examples**	
		Avg	*Min*	*Max*	*Avg*	*Min*	*Max*	*#*	*Avg*	*Min*	*Max*	*Pers.*	*Loc.*	*Obj.*	*Class*	*Avg*	*Min*	*Max*	*Pos*	*Neg*
Short	20	5.02	0.52	22.00	68.5	13	177	55	2.75	1	6	15	10	17	13	32.56	9	60	5.02	1.8
Long	10	77.13	27.85	108.73	606.1	234	889	98	9.8	5	15	31	15	22	28	47.23	21	146	5.35	2.62

Table 5.1: Statistics about the used video datasets: Dataset name, number of videos, duration of the videos, number of extracted keyframes, number of interactively annotated objects, distribution of these annotations over object categories, needed annotation time, and number of average positive and negative examples per annotation (from left to right).

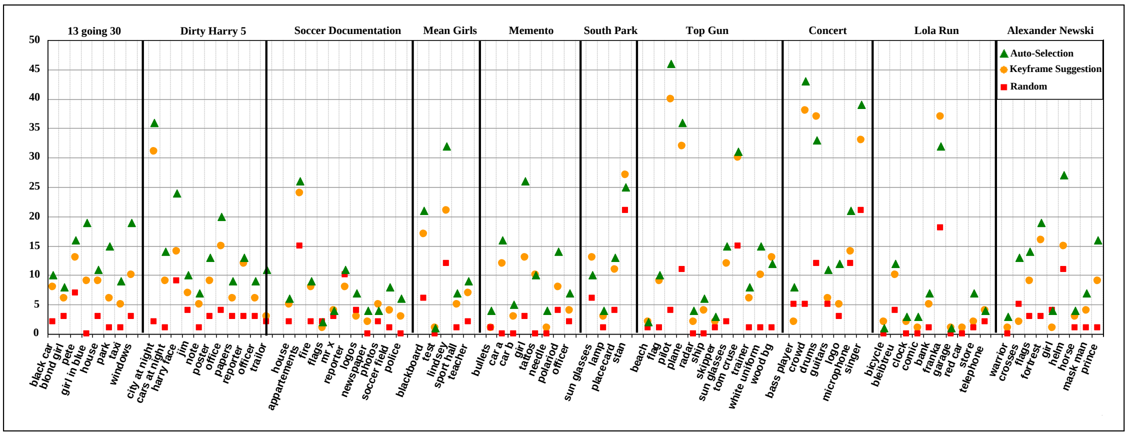


Figure 5.6: Number of keyframes that contain the object-of-interest within 48 keyframes for the short video dataset of professional content, see the video names in the top row of the figure. The results are shown for each object individually using the proposed approach selection (green triangles), keyframe selection (oranges circles), and a random keyframe selection (red squares) as baseline.

After each selection of a positive or negative example, keyframe suggestion is performed on the residual keyframes to support the annotation of further object instances. As shown on the rightmost part of in Table 5.1, more than 5 positive examples were selected in an average annotation, although these statistics include some annotations where only 1 or 2 positive examples were selected. On average the suggested keyframe on position 6.1 was selected as positive example and in 37.5% it was ranked on the first position. It turned out that the selection of the second positive example is the most critical one, especially when a large difference between the initially selected object and all other instances exist. Sometimes, the second selection was placed above the 200$^{\text{th}}$ suggested keyframe. In these cases, the selection of a few negative examples usually improves the situation.

5.3.2 Results

Figure 5.6 and the left diagram of Figure 5.7 show the results of an evaluation that we performed using those interactively annotated object instances of the second dataset that contain more than two positive examples. In these experiments, we manually counted the number of keyframes that contain the object-of-interest (a) within the top 48 suggested keyframes of Algorithm 5.2, (b) within the top 48 keyframes that are returned by the automatically selected recognition approach of Algorithm 5.3, and (c) within a random selection of 48 keyframes from the video. We chose the number of 48 keyframes because this was the amount of visible thumbnails on one screen of the prototype. This experiment setup presents the typical interaction process for user feedback, as described in Section 5.1.2. Note that the randomly selected keyframes have no order and that the ordering (within the 48 keyframes) of the other runs is not important for user feedback as long as all keyframes are shown on one screen because users do not have to investigate the keyframes line by line. Thus, we omit to compute the average precision of correct keyframes and give the absolute numbers of correctly returned keyframes instead. These numbers can be interpreted as precision (using a division by 48) although the actual number of

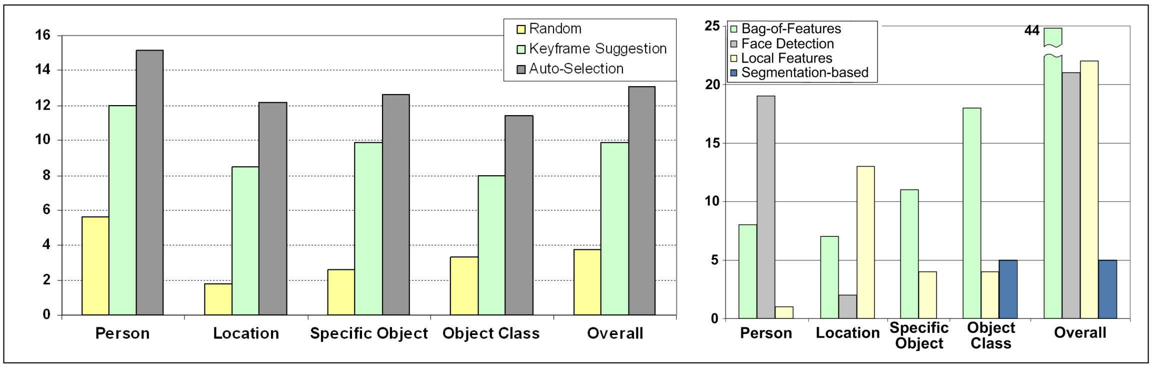


Figure 5.7: Correct keyframes for the three keyframe selection approaches (left) and the selected recognition approaches (right) per object category.

correct keyframes that exist for an object is unknown and only a small fraction of the objects are supposed to be shown in 48 keyframes or more.

The left diagram of Figure 5.7 shows that an average of 10 correct keyframes was returned from keyframe suggestion in the second dataset while almost 13 correct keyframes were detected from the automatically selected recognition approach. The number of correct keyframes from the random keyframe selection was slightly below 4 on average, though up to 21 for the annotation of some persons, as shown in Figure 5.6. However, random keyframe selection performed significantly worse than the object recognition approaches for all object categories and it achieved a slightly better result than keyframe suggestion for less than 6% of all objects. The automatically selected recognition approach outperformed keyframe suggestion in 89% of all annotations and a constant average improvement of about 30% was achieved for all object categories. The right diagram in Figure 5.7 shows the distribution of recognition approaches that make use of bag-of-features (BoF), face detection, local feature matching, and segmentation-based features for each object category. BoF approaches were selected over all categories. For the annotation of persons, face detection approaches (using Gabor wavelets, SIFT, and color features) have been selected most often. Local feature matching approaches, such as the original SIFT matching [191], are the predominant approaches for the location category while segmentation-based approaches (color and texture features from mean-shift regions) were only selected to annotate a few general object classes. In this case study, object regions (Section 5.1.2) have not been used for the recognition approach selection. However, the detected regions are shown in the positive examples of the prototype and they are useful as rough feedback for the quality of the selected examples. If the regions are too big, it is very likely that the object's background is not separately selected as negative example. In examples where the region of wrong objects is shown, negative examples of these objects might be useful. Figure 5.8 shows a few region detection examples.

5.3.3 System Requirements

Last but not least, we measured the processing time and memory consumption of the individual analysis steps on a desktop PC with a Intel quad-core CPU with 2.66 GHz and 8 GB of main

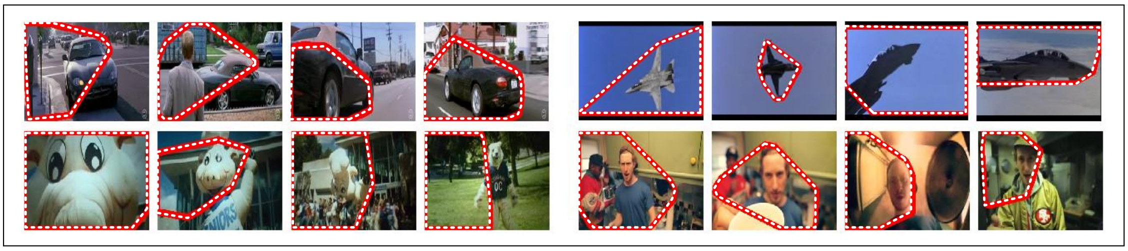


Figure 5.8: Region detection results for 4 objects.

memory. As shown in Table 5.2, the run-times of keyframe clustering heavily depend on the input video size. Videos with a size of 320×240 pixels are clustered 14.8 times faster than real-time, videos with 640×480 pixels 2.7 times faster than real-time, and the clustering of a one hour video with 1280×720 pixels needs about 1 h and 7 min. These run-times include the frame extraction, shot detection, and the actual clustering algorithm.

In the prototype, frame extraction with FFmpeg operates only on a single CPU core and takes between 80% and 90% of the overall clustering time. The clustered keyframes are stored with a resolution of 512 pixels in the longer dimension and consume about 19 MB storage for a one hour video. However, it is not required to store all clustered keyframes permanently, see Section 5.4.1. The keyframe suggestion step needs about 31 s to extract features for 1 h of video while the suggestion of keyframes for a set of positive and negative examples takes only 250 ms. Region detection needs about 750 ms for each keyframe that was selected as positive example while the run-time of the automatic recognition approach selection cannot be given that easily. As shown in Table 5.2, this time depends on the type and number of visual feature and recognition approaches-of-interest. In the case study, an overall number of 12 different feature types with 10 different matching approaches were used, which took between 2 and 3 h to analyze 1 h of video. The size of the metadata that is needed to enable object-based and scene-based video retrieval is below 1 KB for an entire feature film, compare the database schema of Figure 5.2.

5.4 Integration into Video Platforms

In addition to the use of the proposed annotation approach in the prototype of the last section, this approach was designed as interface between video platforms and computer vision research targeting object recognition. We believe, that it offers a good possibility to improve various video sharing platforms (see Section 2.1) in regard to their capability of scene retrieval

Measurement Units	**Keyframe Clustering**			**Keyframe Suggestion**	**Region Detection**	**Auto-Selection**			
	320	***640***	***1280***			***BoF***	***Face Det.***	***Local Feature***	***Seg.-based***
Feature Extraction (sec.)	242	1,315	4,050	31	0.75 /kf	20	125	50	135
Matching (sec.)				0.25		0.2	8	87.5	0.15
Storage (KB)	19,000	19,000	19,000	620	40	445	3,500	15,000	185

Table 5.2: The required run-times and storage for an average video of one hour.

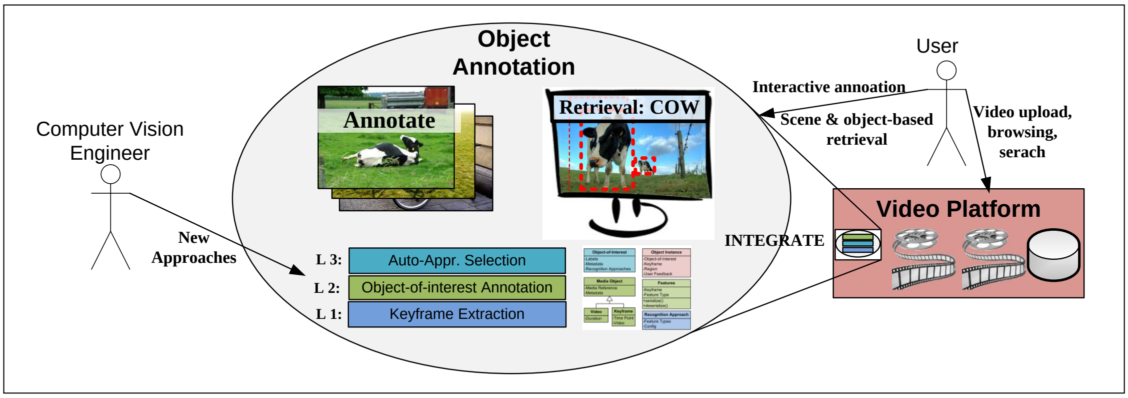


Figure 5.9: A high-level perspective of the roles and integration possibilities of the proposed object-of-interest annotation into existing video platforms.

and object-based text queries, especially as it supports distributed analysis architectures. Figure 5.9 shows a high-level perspective of the approach in combination with the roles of video platforms, their users, and computer vision engineers, as well as an overview of the proposed object-of-interest annotation.

5.4.1 Integration Strategies

There is not only one single way how the proposed framework can be integrated into a video platform, but several integration levels might be reasonable that fit to the available resources and software architectures of different video platforms. In the following, we describe four different integration levels. The first, minimal integration only consists of a manual object annotation that uses the presented keyframe clustering algorithm and the database schema. Thereby, it is sufficient to store one keyframe (or its timepoint) for an annotated object to answer video queries that include the object label as keyword. The second integration level presents an interactive annotation where a video platform has to store all keyframes that a user has selected for an object with the help of the keyframe clustering and keyframe suggestion algorithms. The region detection approach of Section 5.1.2 can be optionally used to support the annotation process. In this integration level, video retrieval became possible on a scene level, which means that users can navigate between different video scenes of an annotated object. The third integration level additionally uses the recognition approach selection of Algorithm 5.3 to annotate the object-of-interest automatically in the first few keyframes that are returned by the selected approach. In this case, it is recommended to store the information if an object instance has been annotated by the user or not and to prefer user-approved instances to answer video queries. The fourth and last integration approach includes an user feedback step to classify the automatically annotated object instances as true or false annotations. The benefit of the later integration levels is a more complete instance list of the object-of-interest and the possibility to (semi-)automatically annotate this object in newly uploaded videos.

Video platforms have to store the annotated objects-of-interest and their instances (compare Figure 5.2) in order to improve the retrieval process. The computation of clustered keyframes and the feature extraction can be done at the client side (user of the video platform) before video upload or at the video platform severs after a video was uploaded. In the case of client side analysis, video platforms do not have to bother about resources and scalability issues. However, we suppose that a stand-alone application is needed in this case instead of a single website. The storage of visual features at the server side is only required if object annotation should be performed in several videos at once.

5.4.2 Discussion

Ideally, the experiments of this chapter would be performed on an existing video platforms but this is out of the project's scope. However, we made some observations in the case study of Section 5.3 that are especially interesting for the applicability of the presented annotation approach in video platforms. Firstly, the objects-of-interest that are suitable for annotation are restricted by their minimum size and by the number of object instances. Only those objects that take at least one third of a keyframe were selected and it takes an exceptionally long time to annotate objects that are only shown in one video scene. Secondly, we found that it is difficult to annotate unknown videos in a way that the selected objects and their labels are not superficial. Thus, we recommend to annotate objects shortly before or after the video upload, see Figure 5.9. Thirdly, we noted it as preferable to select keyframes that show the object-of-interest with the highest amount of variability (e.g. the same person with different clothing, haircuts, and from different viewpoints) instead of simply choosing the first correct suggestion if many positive keyframes are suggested by the prototype.

Another important point for video platforms is the automatic approach selection because, on the one hand, it is known that just one approach is not sufficient for the recognition of all objects. On the other hand, neither the video platforms nor their users can be supposed to select an appropriate recognition approach for each object-of-interest. We have shown that the selected approaches significantly outperform a baseline bag-of-features approach for most examples and that different recognition approaches are selected for different object categories. For instance, face detection approaches are frequently selected for the annotation of persons. It has to be further mentioned, that the author of this work has performed all experiments and annotations in his own right. However, extensible usability studies can be performed by interested readers and video platforms with the annotation prototype, see Section 5.2. This also applies to investigations on changes in the user's retrieval process caused by the existence of new object annotations. Another evaluation that uses the same annotation approach for large-scale video collections is given for the instance search task of the TRECVID challenge in Section 6.1.

5.5 Summary

Despite the fact that object recognition techniques have seen a significant progress during the last years, content-based video analysis is almost not present in today's video annotation and retrieval systems. In this chapter, we propose an extensive annotation approach to enable object-based text queries and video scene retrieval in various systems. The presented annotation process starts by the selection of a single keyframe in which the object-of-interest is shown. Positive and negative examples of this object are then selected by interactive annotation with the help of a fast keyframe suggestion. In a further step, an integrated analysis and evaluation takes place to select an appropriate recognition approach for each object automatically. Eventually, an object is automatically annotated in large-scale video collections and the generated metadata is stored in a compact, binary form. Video platforms can either integrate the entire approach or selected parts of it in order to improve video retrieval on different levels. An advantage of the proposed approach is the systematic support of improved computer vision techniques that will be proposed in the future. High processing performance is further assured by the support of distributed computer architectures and an intelligent reuse of intermediate results during feature extraction.

As a proof-of-concept, we developed a simple annotation prototype and performed a case study on two video datasets. In this process, we demonstrated that the proposed approach can be used to efficiently annotate objects in short videos and full-length feature films. The results of automatically selected recognition approaches significantly outperformed the baseline results. Furthermore, we want to point out that this case study (as well as the evaluations of related works) show that current recognition approaches cannot perform the annotation of any object like the human visual system. However, we believe that the proposed approach will still be useful, if not more effective, when computer vision techniques have improved further. In this context, automatic approach selection will be especially important to support novel recognition approaches that are tailor-made for specific object types. Finally, we measured the annotation time that a user has to spend in the interactive process is 1.5 min for short videos and less than 8 min for entire feature films. We believe that many users are willing to spend such an effort if their videos gain higher visibility as a revenue.

CHAPTER 6

Tasks and Evaluations

This chapter contains further experiments and tasks in the area of video annotation and retrieval techniques that we performed in the context of this thesis. The case study of Section 5.3 reveals that it is very difficult to compare the results of different annotation systems against each other because the output of these systems vary in several matters, such as the representation of objects. Furthermore, users can annotate different objects in the same video, give different labels to the same objects, and operate with a different annotation speed and accuracy. However, these limiting factors do not exist for example-based video object retrieval, and thus initiatives like the TRECVID instance search tasks emerged lately to provide benchmarks for retrieval systems. We took the opportunity to participate at this task in order to compare the proposed object-of-interest annotation against other systems. Thereby, we joint forces with the Institute for Information and Communication Technologies of the applied research company Joanneum Research [147] and participated in three consecutive years (2010, 2011, and 2012). Furthermore, we took part in the TRECVID semantic indexing task where class-level object recognition and the recognition of object-related events are done. As last point of this chapter, we present an action scene annotation based on motion and event detection. The according evaluation combines professional and amateur movies.

6.1 Instance Search

The instance search task exists since 2010 as pilot task with the goal to retrieve video shots of specific objects, persons, or locations. This content-based video retrieval task can be important for many situations and video collections, including archive video search, personalized video organization, surveillance, law enforcement, and the protection of brands [234]. As shown in Figure 6.1, the used object queries are similar to the initially selected objects of the case study in Section 5.3, though a perfect segmentation is given as binary mask. Each query contains between 2 and 6 keyframes of the object-of-interest, extracted from nearby video sequences. As mentioned in Section 2.5, the National Institute of Standards and Technology (NIST) performs the ground-truth generation and the evaluation of this TRECVID task.

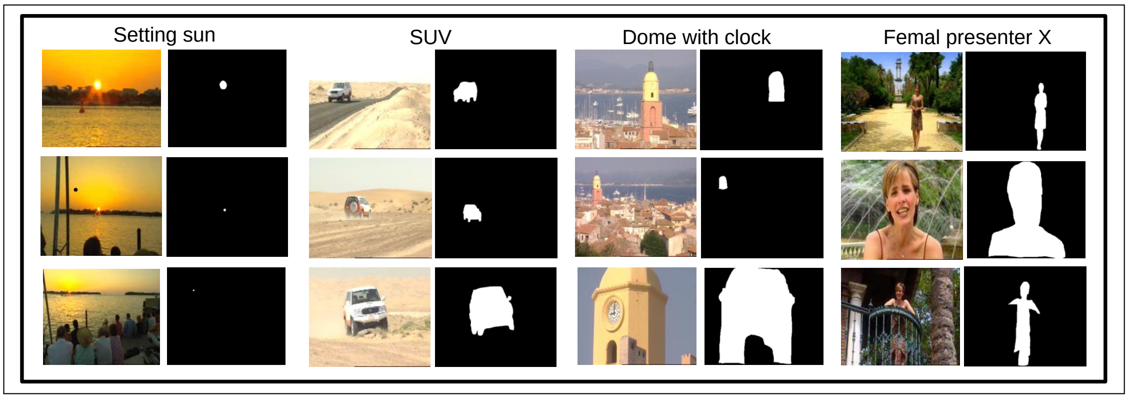


Figure 6.1: Object examples of the instance search task.

In the ground-truth of 2011, 1830 video shots of 25 different objects (see the bottom of Figure 6.3) are annotated in about 136 h of video. The used BBC rushes video collection contains raw material from which programs and films are made in the editing room. These videos include material for several dramatic series as well as for travel programs. In all three years, almost the same number of queries (about 25) is used although the actual queries and the used video material differ. The main difference between 2010 and 2011 is that the latter set contains a lot of almost identical scenes due to the use of the BBC rushes video collection and an additional duplication of all videos with artificial video transforms (change of gamma, contrast, aspect ratio, and hue) in order to simulate the use of different cameras and lighting conditions. In 2012, videos are used from Flickr [94] that are given under the Creative Commons license. Since 2011, NIST performs a shot detection to produce short, roughly equal-length clips (10-20 seconds) and it is specified that the participants have to treat these clips independently during analysis. The output file of each video object query contains up to 1000 ordered clips with those clips on top that are most likely to contain the object-of-interest.

The reason why instance search is still a pilot task is its main intention to explore the task definition and evaluation issues in an approximation to the desired full task using a smaller number of topics, a simpler identification of the target entity, and less accuracy in locating the object instances [235]. In this approximation to the full task, most queries target persons that appeared in different clothes, costumes, and settings, as well as general objects and locations. In the following, we explain the used approach of this thesis and discuss the evaluation of this approach within the 3 participations.

6.1.1 Approach

The first step of the proposed approach selects keyframes from the given clips for further analysis. In 2011, we extracted 69591 keyframes from a total number of 20982 clips using Algorithm 5.1 whereas we simply used every 10th frame in the first year. We then use several subsystems, each performing the content-based search for a certain type of feature with different configurations. In the first year, only the bag-of-features (BoF) approach was used with different configurations whereas a couple of different dissimilarity measures (Canberra distance, Corre-

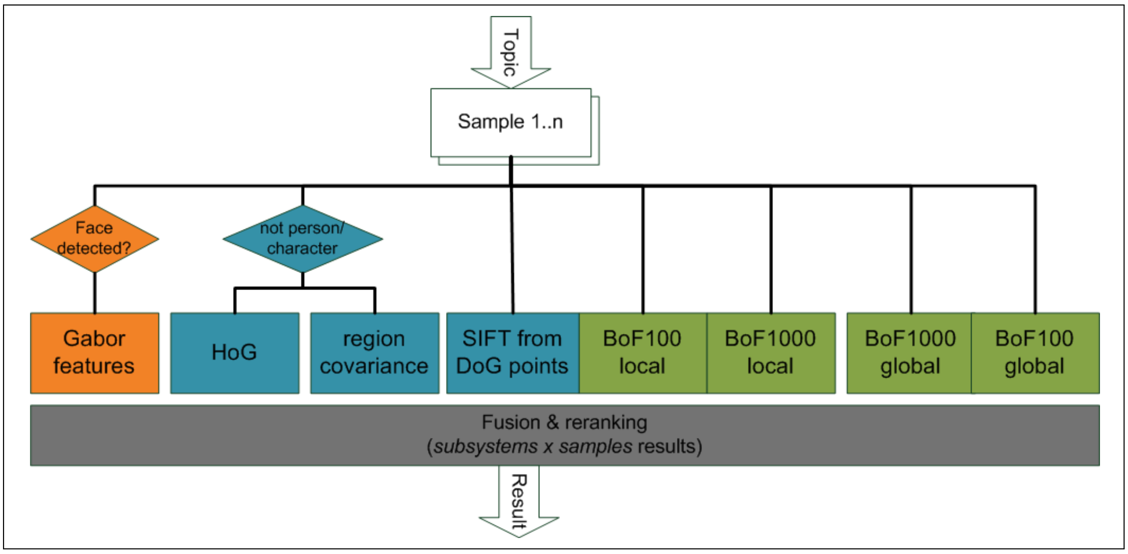


Figure 6.2: Overview of the instance search system from 2010. In 2011, we used a number of different configurations for each subsystem and the automatic approach selection to chose an appropriate setup for each feature.

lation distance, Cosine distance, Jeffrey divergence, Squared distance, ChiSquare statistics, and the Minkowski family distances: Manhattan distance, Euclidian distance, and Fractional distance) are used in the matching step for each feature in 2011. In the latter, we also used the entire query images for feature extraction as well as the cropped objects without background. This leads to a total number of 18 different configurations per feature type. An off-line approach selection was then performed to choose a configuration for each subsystem according to the proposed approach of Section 5.1.3. During analysis each subsystem is queried independently with each sample and returns a ranked result list with a similarity value for each result. We thus have for each query *number of samples x number of subsystems* results. Different fusion methods were used to combine the results of one module (that stem from different images of the same query) and to combine the results of different modules in order to obtain the final result list. Figure 6.2 shows the system of 2010 where Gabor features are used for previously detected face regions and where histogram of oriented gradients (HOG) and region covariance are only used for non-person queries. The differences of the proposed system within the three years account to lessons learned of the previous years, see Section 6.1.2.

Gabor Face Recognition: We perform face detection using the well-known Viola-Jones AdaBoost approach [327] on three scaled versions of each keyframe with the longer image sides of 160, 320, and 640 pixels. Polygon intersection ensures that each face region is not used from multiple scales. After this, every detected face region is processed to generate a 10240 dimensional feature using Gabor wavelets [99]. The Gabor features of all training keyframes are generated and stored in an off-line step. For instance in 2010, this approach leads to a total number of 22856 face features in the database. During instance search, a linear k-nearest neighbor search (see Section 2.4.1) is performed for each detected face in a query image to identify the best matches in the database using the dissimilarity measures mentioned above. For query images where no face was detected, an empty result list is generated.

Bag-of-features (BoF): We generate BoF features using Difference of Gaussian (DoG) and dense sampled interest points, SIFT descriptors, and codebooks that are generated with k-means in an off-line step. In 2012, we additionally use ColorSIFT descriptors [268] and a hierarchical codebook [229] in accordance to the winning approach of 2011 [171], as described in the following experiment section. Moreover, we generate two codebooks with a codebook size of 100 and 1000, respectively. For both codebooks one BoF feature is generated for each keyframe in the test set. BoF matching is mainly performed with the histogram intersection of Section 5.1.2 but, as mentioned before, we investigate other dissimilarity measures too during the automatic approach selection in 2011.

SIFT: A variant of the original SIFT matching approach [191] is used as some kind of baseline in the instance search task. For each query image all DoG interest points inside the object mask and its surrounding are used to extract SIFT descriptors, while we use the entire keyframes of the test clips. In the matching process, each query descriptor is compared with all descriptors of each database keyframe. Then, the top 5 matches are kept for further assessment an a score is assigned to each match using two criteria: the actual similarity distance between the matching descriptors and the consistency of their neighboring keypoints. Thereby, the coordinates of at most 10 neighboring keypoints are projected from the query image to the database keyframe according to the orientation and the scales of both center descriptors. Projected descriptor pairs that achieve a small similarity distance lead to a high matching score for the investigated match.

Histogram of oriented gradients *and* region covariance: We used this two additional features only in 2010 due to their small influence on the overall results. On the one hand, histogram of oriented gradients [59] are state-of-the-art features for the fast detection of pedestrians and have been successfully used for detection of other objects as well. They describe intensity gradients by a histogram of gradient magnitudes in a similar way as it is done for SIFT descriptors. Region covariance descriptors [313], on the other hand, are a representation for rectangular image regions where a covariance matrix is used to describe the correlation of features. This descriptor is robust against noise and illumination changes, and it provides a natural way to combine different feature types, such as pixel color, pixel intensity, image gradients, and the spatial arrangement of the feature points. We used the x and y coordinates, the rgb-color values, and the first order derivatives of the pixel intensity that we calculated with a Sobel-filter.

Mean-Shift segments: In 2011, we performed a mean-shift segmentation to generate about 10 segments per keyframe. We used the mean-shift implementation of [18] that works on a quantized color space. The initial regions of this segmentation are then merged as follows starting from the smallest region. We merge a region with its largest neighbor, until either the number of regions is below 10 or the size of the smallest region's width, height, or area is above 0.02 of the image width, height, or area. After this, one SIFT feature and one ColorLayout feature are extracted from each segment and a k-nearest neighbor search is used again for matching. In 2012, we use this approach to test the semi-local feature extraction of Section 4.1. Thereby, different object-background modifications are applied to the segments before we extract color and texture features from them.

Fusion: As mentioned before, each subsystem generates one output list for each query image. Thus, multiple output lists exist for each query and we have to combine the results of each subsystem first. Next, we have to merge the output lists of different subsystems for the combined runs. In 2010, we experimented with different fusion strategies while we used the fusion of Section 5.1.3 in 2011 for both tasks. The four different fusions of 2010 are based on the similarity scores of each subsystem and the rank of the samples that belong to the same query. Moreover, two weighted fusion approaches use the number of other query samples among the top *k* entries as well as the best ranked other query sample only.

Auto-approach selection: In 2011, we performed four different runs starting with a baseline run using SIFT matching. Moreover, we applied the automatic approach selection of Section 5.1.3 to select (a) the best general configuration of each subsystem for fusion, (b) the best single subsystem for each query, and (c) the best configuration of each subsystem per query for fusion. In the auto-selection process, we compared the results of different sample images of the same query against each other using all subsystem configurations. The idea behind auto-selection is the assumption that recognition approaches that perform well to match the sample images of the same query against each other, are well suited to retrieve further instances of the same object.

Work-sharing: The work-sharing between Joanneum Research and the Vienna University of Technology was done as follows. The subsystems HOG, region covariance, and SIFT matching originate from Joanneum Research whereas all other subsystems have been developed as part of this thesis and they use the proposed recognition infrastructure and object annotation approaches. The fusion strategies of 2010 were implemented by Joanneum Research while it was a part of the automatic approach selection of this thesis in 2011. The 2012 system is based on the one of 2011 with additional features that are proposed in this thesis (see Section 4.1) and by features that were used by the winning approach of the 2011 challenge [171].

6.1.2 Experiments

Obviously, NIST treats the instance search task as a form of search, and evaluates it accordingly with the average precision for each query together with a combined mean average precision for all queries [234]. These measures indicate how much of the annotated video sequences are recognized with a higher weighting of the topmost list entries. Speed and location accuracy are also important but in the pilot version only speed is measured. The runs of each topic are pooled and presented to the human judges of the NIST in order to generate the ground-truth of positive clips. Each participant is allowed to submit 4 different runs in order to test different configurations of their systems. 15 research teams participated in 2010, while 13 groups submitted a total of 37 automatic runs and 4 interactive runs in 2011.

2010: Overall, the top half of the automatic runs had mean average precision (MAP) scores ranging from 0.01 to 0.033. In contrast, the two interactive runs achieved much higher MAP scores of 0.524 and 0.534, and there was a considerable difference in performance from object to object. In this context, it was shown that different approaches were suited best for different query types, and thus there was no approach that performed best for most queries.

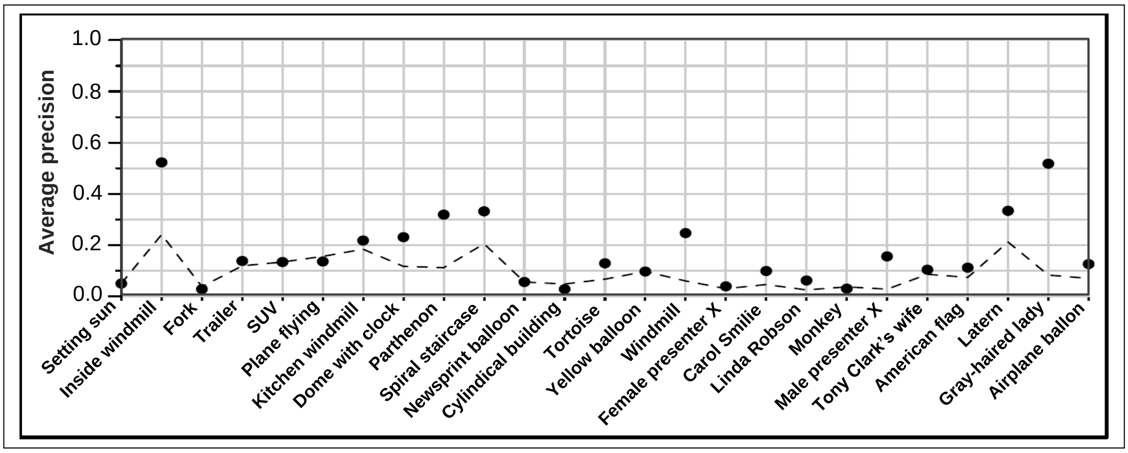
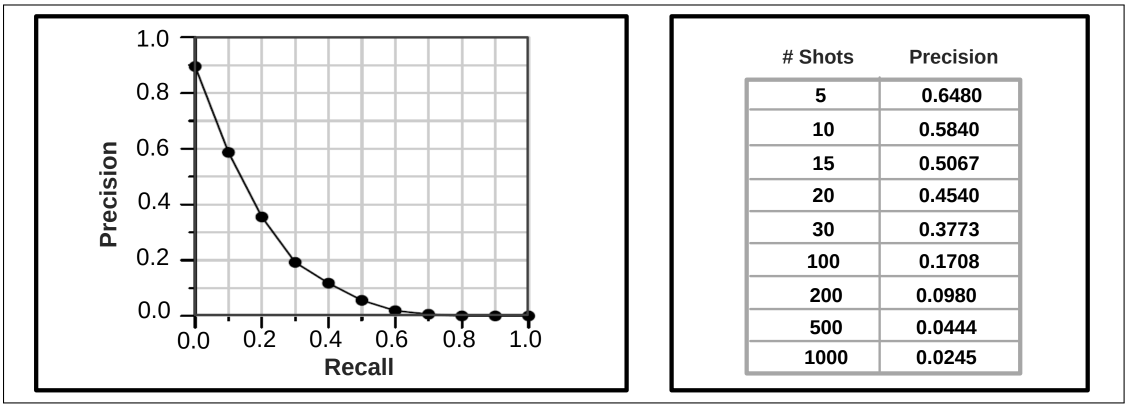


Figure 6.3: Evaluation results for our best run in 2011 (black dots) compared to the median of all participants.

The proposed system of this thesis performed better with weighted fusion methods compared to the simple fusions. A MAP of 0.0066 and 0.0087 was thus given for weighting based fusion in opposite to 0.0051 and 0.0059 for the simple ones. The top run of this system performed slightly worse than the average of all participants (0.0087 to 0.01). We also analyzed the performance of each of the subsystems. For the mean over all queries, the Gabor face feature yields a slightly higher score than the best fused result, although it has not been applied to all the queries. SIFT scores only slightly worse than the best fused results. For person queries the Gabor descriptor scores clearly better than the best fused result, the other features score clearly worse. For object queries SIFT outperforms the best fused result by about 50%. For location queries the best fused result outperforms all single features. These results show that no fusion method is satisfactory, as the weight of the best feature is diminished, so that the best fused result is worse than the best single feature. Thus, we changed our system and fusion strategy for the following participations.

2011: In general, the result of 2011 were significantly better than the ones of the previous year with an increase of the average MAP from 0.01 to 0.1. The best approach [171] achieved a mean average precision of 0.531 and detected 1224 of 1830 relevant clips using ColorSIFT descriptors that are extracted from Harris-Laplace, Difference of Gaussian, and dense sampled interest points. A bag-of-features approach was then applied with a hierarchical pyramid [229] that leads to large descriptors of about 10000 bins and a histogram intersection was used to compare these descriptors. In contrast to 2010 where different approaches performed best for different query types (e.g. face recognition for person queries), this approach performed equally well for all query types. However, the performance was influenced by the object size, as the smallest query objects performed worst. Moreover, the winning team mentioned that their approach performed especially well because the 2011 task was more or less a near duplicate detection task, and that those object instances that appear with a higher variability were not detected. Another mentionable point is it that the best interactive run performed significantly worse than the best automated run.



# Shots	Precision
5	0.6480
10	0.5840
15	0.5067
20	0.4540
30	0.3773
100	0.1708
200	0.0980
500	0.0444
1000	0.0245

Figure 6.4: Interpolated recall-precision curve (left) and precision at *n* shots (right) of our fused system in 2011.

We achieved the best mean precision (0.221) and mean average precision (0.170) with the baseline run of SIFT descriptors. As shown in Figure 6.3, these results are above the median for almost all objects and it is ranked on position 16 of all 37 submitted runs which is a significant improvement compared to 2010. However, the results between the four runs contain some surprises. The runs where only one visual feature was used returned 611 and 613 relevant shots from a total amount of 1830. However, while the baseline run achieved the highest mean precision, the opposite is true for the fused runs. They returned more relevant shots (847 and 783) but their mean precision and mean average precision values are worse than the baseline run. According to the *precision at n shots* statistics, these fused runs further outperform the baseline run between 200 and 500 shots while a much lower precision is achieved in the first 30 shots. The left side of Figure 6.4 shows the interpolated recall-precision curve over all queries and, on the right side of this figure, the precision for the first *n* shots of the output list. About 15% of all object instances are detected with a recall above 50%.

Note that the best run of the challenge (described before) achieved a much higher MAP than our approach while the difference of correctly retrieved clips is not so high (66.9% to 46.3%). The overall evaluation [235] further shows that there is no direct connection between the number of correctly detected clips and the MAP score. The probable reason why the automatic approach selection does not perform as good as assumed is the big difference between the query examples that are used to select the approaches and the actual test clips. While different samples of each query are given with a high variability (e.g. shown from different viewpoints, see Figure 6.1), the video clips in the dataset present something like near duplicates due to the use of BBC rushes video collection and the additional application of artificial video transforms. The runtimes are similarly high in all queries and runs of the proposed system where SIFT matching was involved (between 100 and 300 minutes) while analysis times of a few minutes or even seconds are measured for all other queries. The winning team [171] needed about 15 minutes per query.

6.2 Semantic Indexing

The semantic indexing task deals with the detection of shots that belong to a certain concepts, including class level objects and object-related events. The task exist since 2005, though it was named *high level feature extraction* until 2009. The ability to detect such concepts is an interesting challenge by itself but it gets even more important as fundamental technology for video filtering, categorization, browsing, search, and other video exploitation [235]. Potential applications usually require a large number of diverse concepts, and thus the number of semantic concepts was incrementally increased in the task during the years and relations between the concepts were recently added as additional knowledge source. Moreover, a light version of the task is offered to encourage new participants and teams with lower resources. A secondary goal of the semantic indexing is to encourage generic concept detectors instead of specific detectors for each concept. Due to the common use of modern classification approaches (see Section 2.4.2) this goal is achieved by most participants.

The main differences to the instance search tasks of Section 6.1 are that many training examples are usually given for each concept in the semantic indexing task, and that these examples stem from different video shots. Thus, semantic indexing is considered as classification task by most participants, as they use sophisticated machine learning approaches, see Section 2.4.2. Moreover, real-time requirements are less important than for instance search because the annotation of concepts is usually performed in an offline step and not used for online searches. Finally, the given concepts consider more class-level objects while instance search is done for specific objects.

As mentioned above, the number of concepts increased every year in the task until 500 concepts were selected for the 2011 challenge (130 concepts were used in 2010). A selection of these concepts is given in Table 6.1. These 500 concepts covered as many previous concepts as possible and also comply with many concepts of the LSCOM ontology [225] (Large Scale Concept Ontology for Multimedia). However, only 365 of these concepts were annotated with a large amount of positive examples, and thus used for the experiments. The same concepts are applied in 2012 although additional annotations (described below) are used to generate more examples. Different development sets are further available due to the reuse of concepts over several years. For instance, the entire test data set of 2011 (137327 shots) is contained in the development set of 2012.

In the actual semantic indexing task, participants are asked to return the top 2000 video shots for each concept, ranked according to the internal confidence measures. In this process, the presence of each concept is assumed to be binary, which means that it is either present or absent in a given shot. In 2012, a few pairs of unrelated concepts are further selected by NIST to compare the retrieval of shots that contain a combination of two concepts with individual concept detection. Moreover, the use of development data that consist of positive examples from general purpose search engines, such as Google, is supported by the preparation of an image set for each concept. NIST additionally publishes a master shot segmentation and ASR transcripts of all available data and tries to encourage the sharing of low-level features, detector scores, and entire classification infrastructures between the participants.

Class-level Objects	***Persons***	***Locations***	***Events***
animal	asian-people	cityscape	airplane-flying
vehicle	dark-skinned-people	classroom	bicycling
telephone	female-human-face-closeup	doorway	singing
flowers	old-people	mountain	throwing
boat-ship	hand	indoor	demonstration-or-protest

Table 6.1: Example concepts of the semantic indexing task.

All groups are asked to participate as well in the collaborative annotation to generate a sufficiently large ground-truth, especially for the newly added concepts. An online system is therefore provided that enables the user to annotate shown shots (or their keyframes) as positive, negative, or undecided examples for a given concept. This system applies an ontology and an active learning to generate as much positive examples as possible and to keep negative examples as close to the class boundary (similar but wrong examples) as possible. The active learning work of [15] indicates that this approach achieves the same performance as if the entire training collection is annotated although only a small fraction of carefully chosen samples are actually annotated (about 15 to 20%). The idea behind this active learning is the use of a concept detection system to select the samples that are potentially the most informative ones for training. Compared to early collaborative annotations, TRECVID 2010 to 2012 makes further use of relations between annotated concepts. If, for instance, a shot is labeled as positive example for the concept *adult*, it will be automatically labeled as positive example for *person*; and if a concept is labeled as negative example for *person*, it will be automatically labeled as negative example for the concepts *adult*, *male-person*, *female-person*, and *teenagers* as well. It is worth mentioning that both, the active learning process and the used ontology, are transparent for the annotators in a way that they simply encounter more positive examples than in a full or in a random annotation.

6.2.1 Approach

For the semantic indexing task we use a set of global and local features extracted from dense keyframes and train a classifier for each concept using SVMs. In the following, we briefly describe the used features and kernels before the results are discussed.

Global MPEG-7 features: In all three years, we globally extract the MPEG-7 features ColorLayout, DominantColor, ColorStructure, and EdgeHistograms [199] from the entire keyframes. ColorLayout features describe the spatial distribution of colors with the first low frequency components of a DCT, performed on the average colors of 8x8 pixel regions. Similarly, ColorStructure features capture the spatial arrangement of the colors by counting the number of times a color is present in 8x8 pixel regions. In contrast, the DominantColor consists of up to three representative colors for a keyframe, and we use mean-shift color clustering [18] for extraction. EdgeHistogram represents the spatial distribution of five types of edges, generated from local edge histograms of 4x4 pixel regions.

Global Gabor Energy: We compute the Gabor energy globally by filtering the entire keyframes with a bank of orientation and scale sensitive filters, and by calculating the mean and standard deviation of the filtered outputs in the frequency space. In this process, we apply a fast recursive Gabor filtering for 4 scales and 6 orientations.

Number of faces: As many concepts rely on the existence of humans in a video, we use the popular Viola-Jones face detection [327] to count the number of faces per shot. Shots without any faces are, for instance, supposed to be bad candidates for the person-related concepts given in Table 6.1 whereas concepts like demonstration-or-protest requires a larger number of faces.

Bag-of-features (BoF): In order to use also local features in way that is appropriate for SVMs (with a fixed descriptor dimensionality), we again apply a SIFT-based BoF approach from dense sampled interest regions, compare Section 6.1. Since 2011, we generate different BoF versions where the keyframes are split into 2x2, 1x3, 3x1, and 3x3 regions in horizontal and vertical direction, in addition to a global BoF descriptor. An own 100 dimensional BoF feature is generated for each partition of the splitted BoF versions and they are concatenated to 300, 400, and 900 dimensional features. In 2012, we additionally change the input SIFT descriptors to a ColorSIFT variant where each rgb-color channel is treated independently.

Audio features: In 2012, we started to apply audio features, as some concepts are better described by the audio content of videos than by their visual content (e.g. singing). For other concepts, such as the vehicle classes, audio and visual content can be equally important. The audio features that we investigate are the loudness per frequency band in sone (SONE), Bark Frequency Cepstral Coefficients (BFCC), Mel Frequency Cepstral Coefficients (MFCC), the pitch frequency of an audio frame (PITCH), and the Zero Crossing Rate (ZCR), see [213] for more details.

Classification: As discussed in Section 2.4.2, Support Vector Machines (SVMs) are commonly applied to different classification problems, and they are also often used for concept classification based on low-level features. If we look at the 2009 task [284], all but 3 of the 42 submitters report the use of an SVM variant in some part of their approach. Most of the groups use some low-level features which require other distances than the Euclidean distance between feature vectors, like some MPEG-7 descriptors [199] and variants of histograms. However, only about half of these groups mention the use of specific kernels for these features, while most seem to use the commonly applied radial basis function (RBF) kernel. Thus, we made some experiments with specific kernels in comparison to RBF kernels.

Despite the wide use of MPEG-7 visual features in the research community there is remarkably little work on defining kernels that appropriately model the proposed distance functions. A kernel combining different MPEG-7 features and considering the appropriate distance functions was proposed in [71, 72] and it performed better on a small image data set. We thus define a kernel that combines appropriate kernels for the different features:

$$\kappa - combined(x, x') = \kappa - mpeg7(x, x')\kappa - bof(x, x'), \tag{6.1}$$

where $\kappa - mpeg7$ is the kernel for MPEG-7 features described in [19]:

$$\kappa - mpeg7(x, x') = \prod -i \in \{cld, dcd, csd, ehd\} \exp(-\bar{w} - i\kappa - i(x, x')). \tag{6.2}$$

The feature weights $w - i$ are defined as:

$$w - i(T) = \frac{\mathrm{var}(\{d - i(x^- - i, y^- - i) | \forall x^-, y^- \in T^-\})}{\mathrm{var}(\{d - i(x^+ - i, y^+ - i) | \forall x^+, y^+ \in T^+\})}, \tag{6.3}$$

where x^+ (x^-) denotes a positive (negative) sample in the training set T, and $d - i(\cdot)$ is the distance function for feature i. The weights are defined as the ratio of the variances of the feature distances among the negative and positive samples. The weights for the individual features are then normalized to obtain $\bar{w} - i = \frac{w-i}{\sum -j \in \{cld,dcd,csd,ehd\} w-j}$. In contrast to [72], we calculate the weights in advance and not iteratively during training.

$\kappa - Gabor$ and $\kappa - nrFaces$ are RBF kernels while the BoF descriptors are classified with a histogram intersection kernel $\kappa - bof$ according to the BoF-matching of Section 5.1.2. The kernel is defined as:

$$\kappa - bof(x, x') = \sum -j = 1^n \min(x - i, x' - i), \tag{6.4}$$

with n being the size of the BoF vocabulary. The audio features of 2012 are classified with a linear kernel.

Semantic Relations: As mentioned in the task description, a set of simple semantic relations is given for the concepts of the LSCOM ontology and we used them as a post-processing step in the 2011 challenge. On the one hand, *implies* relations are given to state that the existence of one concept automatically means that other concepts are also given. For instance, *female-faces* imply the concept *person*. On the other hand, *excludes* relations mean that two concepts cannot exist together in one shot, like *indoor* and *outdoor*. In the experiments, we first resolve possible transitive relations and added them to the set of relations. We then applied the following rules to improve the final predictions for those concepts that are given in the semantic relations.

Consistent predictions for concepts A and B:

$$p(B) = \max(\min(p(B) + 0.5(p(A) - 0.5), 1.0), 0.0) \tag{6.5}$$

Contradicting predictions of A and B: We used three options: (a) optimizing recall, i.e. enforcing the relation, (b) optimizing precision, i.e. removing the prediction that corresponds to the condition in the relation, and (c) deciding based on the relative margin of the two predictions to the decision boundary. There are cases where two relations have the same target, so that potentially concurrent updates could happen. We therefore used different strategies to combine these updates as their minimum, maximum, mean, or median. However, no differences in terms of the results have been observed.

Work-sharing: The semantic indexing task is only roughly related to the object-of-interest annotation work of this thesis. Thus, we only participated to the submission by providing BoF features and the number of faces that are similarly used in the instance search task of Section 6.1. The spatial pyramid BoF and the ColorSIFT version were added in 2012 due to their good results for related class-level object recognition tasks [268]. The audio features also account to the Vienna University of Technology, though they are no direct part of this thesis. Keyframe extraction, global MPEG-7 features, as well as the classification and fusion was performed by

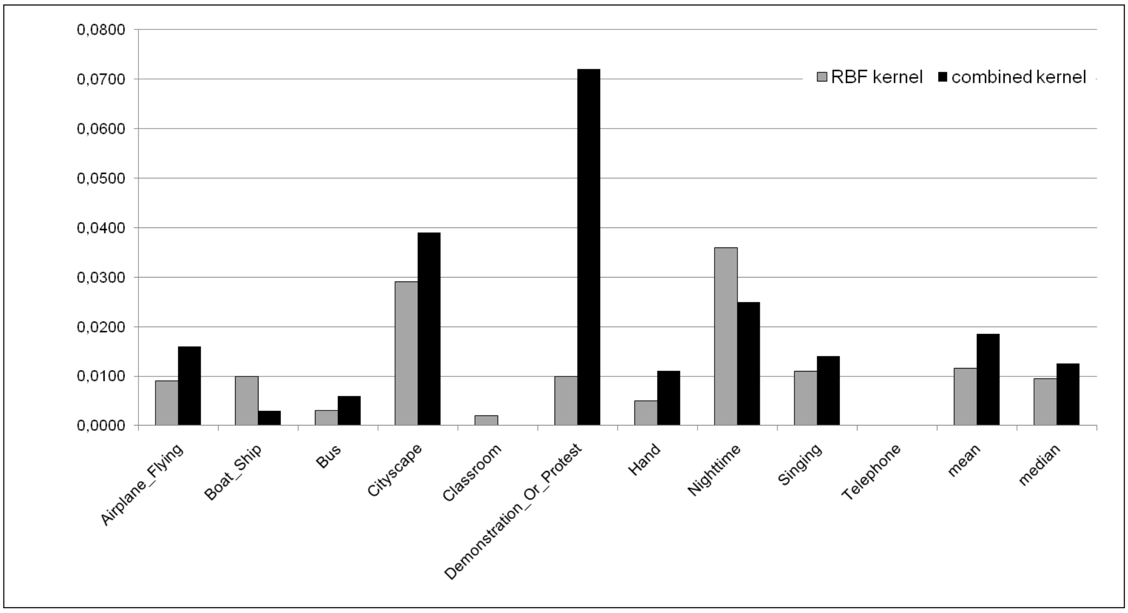


Figure 6.5: The semantic indexing results of the two proposed kernel variants in 2010.

Joanneum Research. However, we participated at the collaborative annotation and held regular meetings with the Joanneum team to jointly decide which experiments we should conduct.

6.2.2 Experiments

As mentioned in the introduction, each group can submit up to 4 full run and light runs with a reduced number of concepts every year. The 2012 challenge provides another opportunity to submit 2 runs for the selected concept pairs. The number of participants between 2010 and 2011 significantly decreased considering both, participants (69 to 56) and finisher (39 to 28), possibly because a new multimedia event detection task was introduced in 2011. We only submitted light runs in all participations due to the immense efforts that are required for the full runs. This light runs contained 30 concepts in 2010 and 50 in the two following years.

Similar to the instance search task of Section 6.1, NIST performs the evaluation with an software tool to calculate the inferred recall, inferred precision, and the inferred average precision. Since all runs provide results for all evaluated concepts, runs can be compared in terms of the mean inferred average precision across all evaluated concepts [14]. The inferred average precision (infAP) estimates the average precision surprisingly well using a small sample of manual inspections from the submitted shots of a concept. Thus, more concepts can be evaluated by NIST with the same efforts [234]. During evaluation, the top ranked video shots (usually ranks 1 to 100) of each group are watched completely by human judges while listening to the audio. Although participants of both semantic indexing variants have to submit results for all selected concepts (365 or 50), NIST only evaluates a subset of it (50 concepts for the full run and 10 concepts for the light run) due to the required judgment efforts.

2010: The overall performance of all participants varied greatly by concept. The inferred true positives (TPs) of 13 concepts exceeded 1% TPs from the total test shots. The concepts *vehicle* and *ground-vehicle* had TPs in over 3% of the test shots. Moreover, some concepts show a large spread between the scores of the top 10 groups (animal, bicycling, singing, and demonstration-or-protest). In general, the median scores ranged between 0.001 (sitting down) and 0.117 (swimming). The median scores of all groups were above a random baseline run except for 8 concepts, for which the random and median values were very close as these concepts achieved to lowest median scores. As a general observation, the top 10 performances for the majority of the common features were less than the top 10 scores for 2009. This was probably due to the high variation in the 2010 data.

We have submitted 4 runs, using two different kernels and we train the system on the two different development sets of 2010 and 2007. On the one hand, our hypothesis for the differences between the kernels was that the combined kernel of different kernels per features should perform better than the RBF kernel that treats all the input features identically. As shown in Figure 6.5, the combined kernel really outperforms the RBF kernel for 6 out of 10 concepts in terms of infAP, and yields also better mean and median infAP over the 10 concepts. However, the strong improvement for demonstration-or-protest seems to be an outlier while the concept telephone seems not to be represented by any of the used low-level features. The results of the combined kernel were only slightly below the median of all participants (0.0019 to 0.0021).

On the other hand, we investigated the use of different development sets with the 2007 and 2010 data sets, at least for the three concepts that they had in common (bus, boat or ship, demonstration-or-protest). Unfortunately, we cannot draw a general conclusion from the achieved results: For boat-or-ship the combined kernel scored worse than the RBF kernel on the 2010 data against the overall trend, while with training on the 2007 data the results for both kernels are the same. The concept demonstration-or-protest has slightly changed from people-marching in 2007, which might be one reason for the decrease in performance. In conclusion, the results show that it makes sense to use appropriate kernel functions for the features involved instead of the *off the shelf* RBF kernel. Concerning the generalization properties of the kernel across data sets it is unfortunately not possible to draw clear conclusions.

2011: As shown in Figure 6.6, the frequency of hits varies again heavily in 2011 by concept. The concepts *adults*, *indoor*, *male-person*, and *text* has inferred TPs in over 10% of the test shots. On the other extreme, the concepts *charts*, *people-marching*, *sitting-down*, and *door-opening* had a TPs in less than 0.3% of these shots. A general observation in this context is that the top performing concepts are more generic by their definition than the bottom performing ones that are very specific, such as *sitting-down*. As in 2010, some concepts showed a large spread between the scores of the top 10 performers (beach, charts, news, singing, and skating).

In general, the results of the task improved significantly from 2010 to 2011 by an approximated factor of 2. The highest mean infAP was about 0.17 (compared to 0.09 in 2010), and a similar high score was achieved by several groups for the full run. The median of all teams was 0.109. In contrast, a median of only 0.056 is given for the light run although the top mean infAP is similar high than in the full run with 0.149, achieved by the same teams. The median scores furthermore ranged between 0.002 (sitting down) and 0.441 (studio-with-anchor-person). Additional observations [235] are that too much time was spent on feature extraction instead

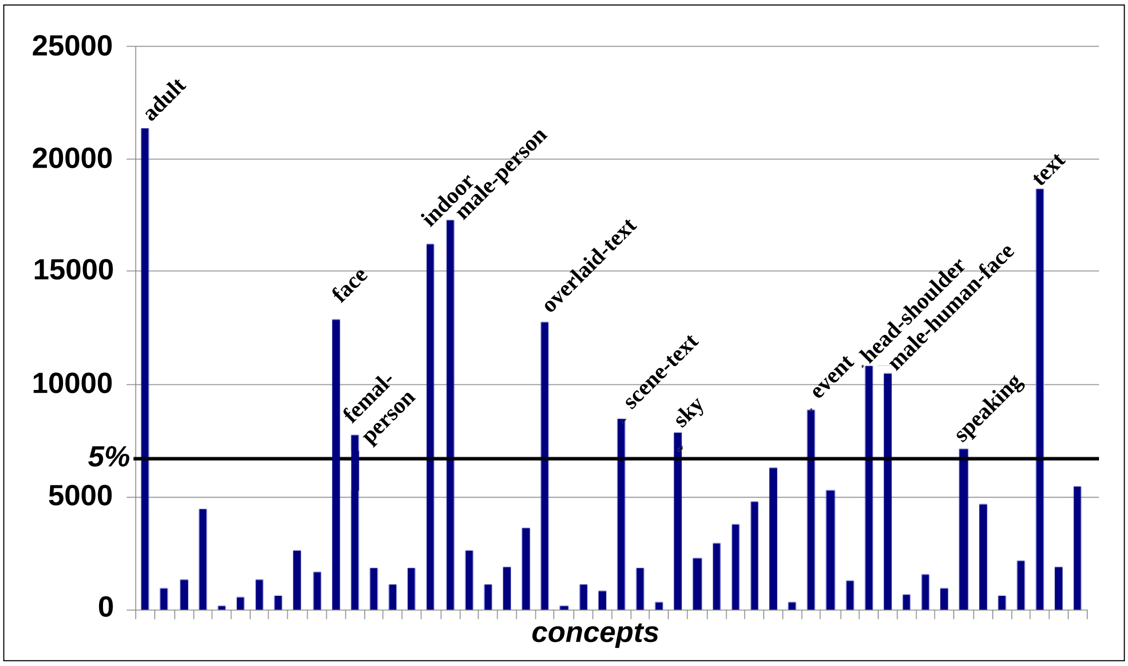


Figure 6.6: The total inferred true positive hits per semantic concept in 2011. The 5% line marks those concepts that are containt in more than 7340 shots of the 146788 shots of the test set.

of developing new frameworks and learning methods. Moreover, the computational power and storage capacities played an important role to get promising results, and there was a learning problem for those concepts that only consists of a few positive examples in the development set. In general, systems focused on robustness, to merge many different descriptors, most made use of spatial pyramids and improved bag-of-features approaches, as well as improved kernel classification and sophisticated fusion strategies. Some teams experimented with the analysis of more than one keyframe per shot whereas only a few teams made use of motion information. Only a few experiments used external training data and no improvements were shown by them.

The top performing teams generally made use of a large number of different descriptors and classifier variants. For instance, [266] applied color histograms, Gabor and quaternionic wavelets, SIFT, ColorSIFT, STIP, percepts, and MFCC audio features in combination with different quantization and spatial alignment methods (resulting in a total of 47 descriptor variants). In addition to k-nn and boosted multi-SVM classification, this group further applied a temporal re-ranking of shots that belong to the same video. In their four runs, they found out that the use of many additional descriptors and processing steps that all lead to a modest improvement alone can result in a significant improvement of the global performance.

Due to the large resource and computational efforts that are required in the instance search task, we were only able to use a subset of the features that we extracted (mainly the MPEG-7 features) while a lot of other participating teams did not manage to submit a single run at all. In this way, we were satisfied with the Olympic idea that taking part is everything, and focused on a few research questions. On the one hand, we wanted to try a different number of positive and negative examples to train concepts. On the other hand, we investigated if the provided concept relations improve the overall results. Although the proposed system did not achieved competitive results with respect to the infAP (see Table 6.2), the results has lead to a surprisingly high number of unique shots that are not detected by any other group. In the light runs we detected 404 of these true shots which is the second rank of the 23 participants and this amount would also be ranked on the sixth position in the full runs. A reasonable explanation for this success was the use of global MPEG-7 features that are not part of the best practice nowadays.

Table 6.2 shows an overview of our runs and their parameters, as well as the achieved mean infAP. As shown in first two lines, training on the same number of samples of the TV10 development set only yields slightly worse results than training from the TV11 set. Note that this TV11 set combines the TV10 data with the newly annotated set of 2011. There is an improvement of the results with increasing number of samples from the training set. However, the improvement is only moderate. On the other hand, the use of semantic relations did not improve the results at all. Some concepts were unaffected, for others the results got worse. It seems that the probabilities generated by the SVM are not appropriate to compare and adjust the reliabilities of the different concepts in a post-processing step.

6.3 Action Scene Annotation

The detection of video scenes that contain specific content or events is interesting for various applications, like video search, summarization, classification, and navigation. Content-based analysis methods for automatic scene detection mainly differ in the scene types-of-interest and the analyzed content. This section investigates approaches to detect action scenes in feature films and user-generated video. In contrast to the generalized concept detection of the last section, tailor-made features and event detection strategies were therefore developed in this thesis with the automatic approach selection of Chapter 3. The proposed techniques differ from existing action scene detection works as they are the only ones that are similarly suitable for professional and amateur content.

Well-defined rules exist for feature films to classify scenes by considering entities like time, place, and story line. One rule states that action scenes contain a series of shots with high motion activity and fast edits. This creates tense atmosphere and a sense of kinetic action and

training set	***# of samples***	***relations***	***infAP***
TV11	500	no	0.0102
TV10	500	no	0.0079
TV11	1000	no	0.0105
TV11	1000	yes, based on run 4 (relative)	0.0043
TV11	1000	yes, based on run 4 (precision)	0.0040
TV11	1000	yes, based on run 4 (recall)	0.0040
TV11	1000	yes, based on run 6 (relative)	0.0047
TV11	5000	no	0.0118

Table 6.2: Overview of the semantic indexing runs in 2011.

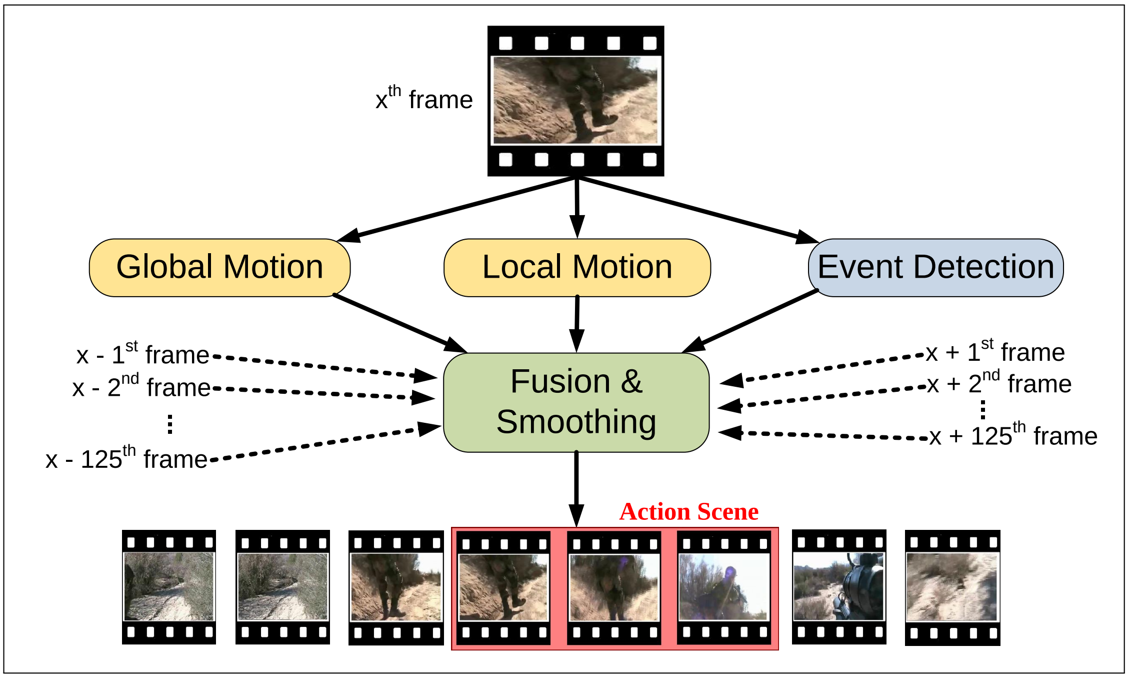


Figure 6.7: Overview of the proposed action scene detection approach. Global motion, local motion, and event detection are first used to classify each video frame independently. A fusion and smoothing step is then performed to generate continuous action scenes.

speed [32]. However, such rules usually do not apply to user-generated content where action scenes have to be formulated in a more intuitive way. Therefore, we just suppose that action scenes contain specific events, such as explosions, car crashes, and gun shoots, in opposite to non-action scenes.

Existing scene classification systems roughly apply the following approach: (a) use shot boundary detection (b) to segment different scenes by a clustering of similar, nearby shots and (c) assign each scene to one of the given classes. In opposite to such approaches, we investigate if action scene detection is possible without shot and scene segmentation to gain following benefits. On the one hand, this approach is more flexible as it allows the beginning and end of scenes to lie within shots instead of forcing them to be on shot boundaries. Thus, it is especially interesting for user-generated videos with no or only few shot boundaries. On the other hand, it can be used to classify very short video sequences with a length of just a few seconds. The proposed approach works on video frame basis where each frame is classified independently by global motion, local motion, and visual event detection. A fusion and smoothing step is then performed to generate larger video segments that are classified as action or non-action scenes, see Figure 6.7. In this process, motion detection can be seen as the foundation of the approach while event detection is used to enhance the results. In the following, all steps are explained in detail.

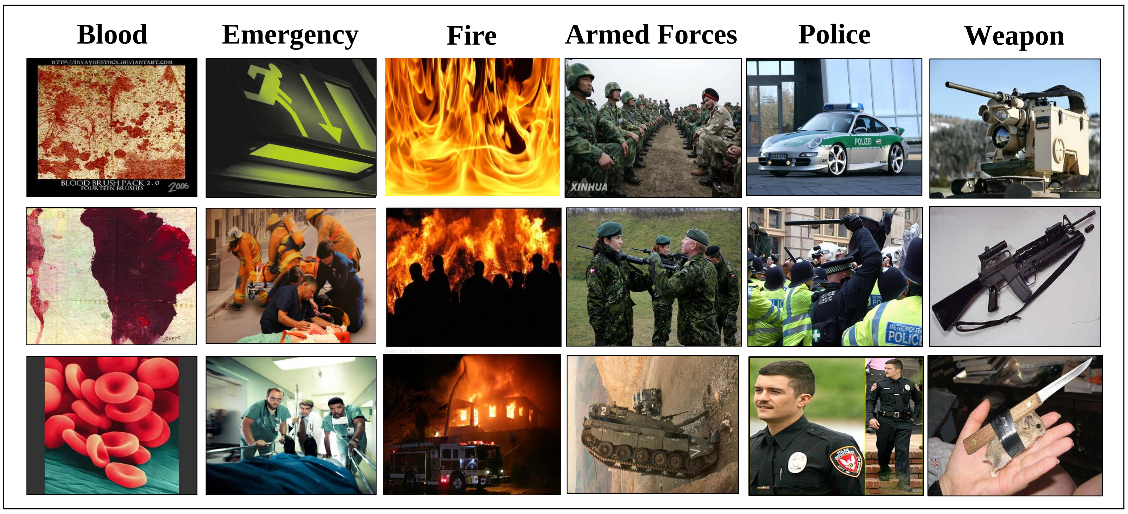


Figure 6.8: Example images that are selected for event detection.

6.3.1 Approach

Motion detection: We use the global and local motion features described in Section 4.3. The classification of each frame is done individually. Thereby, the respective motion features represent the input for SVM classification with a radial basis function (RBF) kernel that was chosen as in [48]. The parameters of the kernel are optimized with a grid search strategy, see Section 2.4.2. Training is performed with a set of 100 features from action and non-action sequences, extracted from one movie of the test set.

Event detection: We use state-of-the-art object classification methods to find a set of events that indicate violent activities and action scenes. These events are represented by images labeled with *blood*, *emergency*, *fire*, *armed forces*, *police* or, *weapons* where event-related objects are shown, see Figure 6.8. These images have been downloaded from Google image search and Flickr for each of these labels and for an additional non-action event class and again about 100 images per class are used for training. Event detection is then performed with the a bag-of-features approach using MPEG-7 ColorLayout features that are densely sampled from about 300 uniform image regions considering 3 different scales, in accordance to the similar keyframe detection of Section 5.1.2. The ColorLayout features have been automatically selected because the events-of-interest seem to be better described by color than by texture or shape.

All ColorLayout features extracted from the downloaded images are clustered with k-means to generate a codebook. A codebook dimension of 250 was chosen because of the faster computation and similar results compared to the dimensions 500 and 1000 on a single test movie. For training and prediction ColorLayout features are extracted from images of each class and from each video frame, respectively. Finally, each ColorLayout votes for up to 3 codebook bins (cluster centers) using a nearest neighbor search with Euclidian distance and a distance threshold of 0.5. Classification of each input frame is then done with a cascaded SVM with RBF kernel, where the 6 event classes as well as the non-action event class are learned. The best matching class together with a posterior probability estimate are used as classification output.

Fusion: Motion classification provides the basis for the combined results. If the global and local motion results agree on the same class for a frame, this result is used without considering the results of event detection. Otherwise, the event detection results decide if a frame belongs to the action or non-action class. We propose an alternative fusion for applications where only global motion or local motion is used in combination with event detection. In this case, event detection overrules the motion classification of those frames where the posterior probability estimate of an event class is higher than 80%.

Smoothing: As last step, we smooth the fused results to generate longer sequences of the same scene type. Thereby, a frame is classified as action or non-action scene when at least 150 of the next 250 frames vote for action or non-action, respectively. Frames are not classified when fewer votes are given for both classes which happened to be the case for less than 5% of all frames in the following experiments. The size of 250 frames was chosen according to the minimal action sequence length (10 seconds) in the ground-truth.

6.3.2 Experiments

We have evaluated the proposed system on two different datasets, one with Hollywood action movies and another, smaller one, with user-generated action movies. Furthermore, the proposed approaches have been evaluated on frame-level and scene-level to allow for better comparison with existing approaches.

Dataset: The first dataset includes the 10 feature films listed in Table 6.3. The length, number of action scenes, and percentage of action frames of these movies are also given in the table. The second dataset also consists of 10 action movies but generated by amateur filmmakers. These movies were taken from YouTube and have a length between 4 and 11 minutes. Although most of these movies follow a straight story line and consist of different shots, there are considerable differences to the movies of the first dataset. For instance, bad lighting and stabilization effects often occur together with video artifacts, only inexpensive special effects are employed, and toy guns with imitated shot sounds are frequently used. In order to generate the ground-truth for both datasets we followed the approach of [174] and annotated scenes as action scenes when at least one of the following events occurs: fire or explosion; violence like fighting, gun shots, robberies, shouts and screams, car chases or crashes, and sounds like alarm or breaking glass. Since there are no open evaluation sets for action scene detection so far, we made this publicly available under [1].

Results: The columns in the middle of Table 6.3 show the frame-level results of the individual modules for the Hollywood dataset. The recall values (left sub-columns) state the percentage of correctly classified frames while the precision values (right sub-columns) state the number of correctly classified frames divided by the number of all classified frames. With an overall recall of 64.2% and a precision of 74.3% global motion performs only slightly worse than local motion. The overall precision of the event detection module, 67.8%, is also promising although its recall is only 22.3%. This low recall follows the fact that only those frames (32.9%) with a posterior probability estimate above 80% are classified as action or non-action scenes for this evaluation.

The results of the combined system are shown in the rightmost columns of Table 6.3 for a frame-level and a scene-level evaluation. On frame-level, the combined system achieved a much higher recall than all individual modules (+5% compared to local motion) and a slightly higher precision. The scene-level results are computed according to [174]. Although the overall results of both evaluations are similar, there exist high variations for some movies. Especially Movie 4 (The Hunted) and Movie 8 (Smokin Aces) have much better results for the frame-based evaluation than for the scene-based one. The main reason for this is the low action frame percentage of these movies (see column *# Action Frames*) as correctly classified non-action frames count higher for the frame-based evaluation. The achieved recall of 70,6% is lower than the recall of the related shot-based systems of [48,106,174,187] (74% - 96%) while the precision of 70.9% is similar or better (62% - 71%). However, most of these related systems are evaluated on smaller datasets of only 4 movies.

A further frame-level evaluation was done for the amateur action movie dataset. The achieved overall recall of 68.2% and precision of 71.6% are just a few percent lower than for professional material. A closer examination of the separated motion and event detectors indicates that local motion and event detection works fine for this content while the global motion performance decreases compared to the results of Table 6.3. We conclude that large motion changes are captured from unstable handheld cameras even for scenes without significant movement. However, as the dataset size is very small with a total length of just 67 minutes it is unclear if these results can be generalized to different kinds of user generated content.

All videos of the evaluation were given with a frame rate of 25 fps and a frame size of 480x320 was used. The motion and event detectors were developed in C++ and with the recognition infrastructure of Chapter 3. The performance evaluation was done on a single core of an Intel 2.66GHz quad core machine. Global motion achieved an analysis rate of 125 fps whereas only 4 and 14 frames were analyzed per second by the local motion and event detection modules, respectively. These values include image IO as well as the SVM classification.

	Statistics			**Frame-Level Evaluation**								**Scene-Level**	
Film Title	***Length*** (*min*)	***# Action Frames***	***# Action Scenes***	***Global Motion***		***Local Motion***		***Event Detection***		***Fusion & Smoothing***		***Fusion & Smoothing***	
Crank 1	84	33%	15	58.1	67.4	63.6	71.9	16.1	67.5	63.8	70.0	66.7	68.8
Crank 2	96	47%	26	58.2	67.7	59.8	66.8	15.0	71.6	63.3	68.8	88.5	71.9
Boondock St.	108	35%	23	59.9	68.7	61.5	67.6	14.8	73.5	65.8	68.9	69.6	69.6
The Hunted	91	23%	19	69.8	80.0	71.8	78.7	24.5	84.9	78.9	83.5	63.2	63.2
John Rambo	87	30%	20	67.6	77.9	68.5	77.2	25.3	70.9	74.5	79.0	77.3	68.0
Public Enem.	143	18%	16	71.1	82.0	75.9	83.1	33.1	77.0	82.6	87.1	93.8	83.4
Shoot Em Up	87	48%	19	61.0	70.9	61.3	68.9	31.8	56.3	63.7	70.2	57.9	68.8
Smokin Aces	104	22%	23	67.7	75.4	73.2	79.0	32.9	74.4	78.7	82.1	56.6	68.4
Transporter	95	34%	13	56.5	72.0	62.5	75.3	12.1	58.9	61.7	76.5	69.3	64.3
Wanted	95	42%	23	62.5	74.3	64.9	71.8	12.8	40.2	71.8	71.3	62.6	72.0
Overall	**972**	**33%**	**198**	**64.2**	**74.3**	**67.9**	**75.0**	**22.3**	**67.8**	**71.6**	**76.7**	**70.6**	**70.9**

Table 6.3: Statistics about the professional video data set and the achieved results. Frame-level and scene-level evaluations give the recall (left sub-columns) and precision (right sub-columns) for individual modules and for the fused and smoothed results.

6.4 Summary

The presented annotation approach of Chapter 5 achieved competitive results for example-based video object retrieval in the instance search task of the TRECVID challenge. These results support the approach's usability for video portals as a similar number of correct object instances were detected under the topmost shots although a much larger dataset was used. The achieved results are above the median of all participating teams though they are not the best overall results. However, we do not intend to prove that this approach would work better than all other systems because further recognition approaches (including the ones that performed best in the challenge) can easily be integrated. Moreover, we participated at the semantic indexing task of the TRECVID challenge where class level object and object-related events are given as concepts. In this task, we focused on specific research questions, such as the use of concept relations and different development data sets, instead of aiming at a competitive overall system. Note that the members of Joanneum Research are the leading force of this joint task participation whereas it is only of minor interest for this thesis.

A further contribution of this chapter is an action scene annotation system that works fine without using the underlying video structures of feature films, namely shots and scenes. For this task, we propose the combination of several state-of-the-art content analysis approaches that lead to different trade-offs between accuracy and speed. The proposed global motion approach appears to be the best general choice considering the high frame rate and the good results. The overall system shows similar results as related shot-based approaches on professional feature films and promising results on user-generated action movies. In order to facilitate reproducible research, we further made the ground-truth of our test set publicly available [1] and we participated at the collaborative annotation of the semantic indexing task as well.

CHAPTER 7

Conclusions

Intelligent video annotation and retrieval techniques are required to use video more effectively in many means. This thesis investigated such techniques, and a brief summary of its outcome as well as an outlook are given in the following.

7.1 Summary

Despite the fact that object recognition techniques have seen a significant progress during the last years, content-based video analysis is almost not present in current video annotation and retrieval systems. Thus, the main research question of this thesis is: *How to employ object recognition for intelligent video annotation and retrieval tools?* In order to answer this question, we first present the state-of-the-art of current video annotation and retrieval systems, the visual analysis techniques that might be used to improve these systems, and related research directions. In addition to summing-up a wide area of research, we incorporate video platforms, search engines, and annotation systems that are available on the web and discuss the requirements and intentions of different users.

In this thesis, we propose an extensive annotation approach that is meant for the use in various video platforms to enable object-based text queries and video scene retrieval. This object-of-interest annotation starts from a single object example given by a user. The approach is easy to use and its main advantage is the systematic support of improved computer vision techniques that will be proposed in the future. Moreover, a recognition approach is automatically selected for each object-of-interest by an integrated analysis and evaluation process. Eventually, an object is annotated in large-scale video collections and these annotations are stored in a compact, binary form together with the selected recognition approaches and low-level features. The foundation of this flexible approach is CORI, a configurable object recognition infrastructure that is suitable to develop recognition applications from simple configurations without high development efforts. Instead, a simple configuration is sufficient that can be generated easily, even from developers with little computer vision background. CORI can be extended with all kinds of visual features, training and recognition strategies, and high analysis performance is assured by the support of distributed computer architectures and an intelligent reuse of intermediate results within automatically constructed feature extraction graphs.

As mentioned above, CORI facilitates the selection and customization of recognition approaches for complex task specifications that can hardly be achieved otherwise. The entire recognition process is thereby investigated to select an appropriate setup amongst thousands of possibilities for a given task, domain, or dataset. In contrast to this holistic approach, related works optimize only specific components of the recognition process.

As proof-of-concept it was shown that the presented object-of-interest annotation approach works efficiently for the instance search task of the TRECVID challenge and for an extensive case study that we performed on a prototype. The achieved results and the measured system requirements indicate that the approach can be used to facilitate video platforms right now. Moreover, we participated at a second TRECVID task and presented an action scene detection system that works fine without using the underlying video structures of professional and amateur movies. Again, both evaluations lead to promising results although we did not achieve the highest recognition results in these challenges. However, we do not intend to prove that the proposed approaches would work better than all other systems because further recognition approaches (including the ones that performed best in these challenges) can easily be integrated due to the use of CORI.

In addition to contributions for video annotation and retrieval, we developed a set of novel visual features and feature extraction techniques. Semi-local features present one of the first segmentation-specific feature extraction methods that is compatible with many state-of-the-art descriptors. Furthermore, we used local feature matching to select object regions that are repeatedly shown in nearby video frames and similarly annotated images. Two novel motion features that capture activity changes of short video segments in a very compact form are finally given. A special focus of this thesis was further set to achieve reproducible research. Thus, we made the proposed video object annotation prototype freely available. Open datasets and challenges were used for the evaluations whenever possible and the newly generated datasets are published as well. In addition, some of the resulting publications of this thesis are published as open access variant.

7.2 Outlook

The current state-of-the-art leaves a lot of space for further advances and research in video annotation and retrieval systems. Open points include innovative representations of the retrieval results, browsing facilities to navigate from one scene-of-interest to the next, and the definition of object-based search queries. The used annotation systems and the generated metadata play a central role in this context, as temporal and spatio-temporal annotations are a prerequisite to enable object-based and scene-based video retrieval. Content-based video retrieval, on the other hand, is mainly interesting for copy-detection issues and its use as refinement strategy for text queries.

With the demand of mobile applications, research for location-aware video annotation and retrieval further rises. The retrieval of videos that are shot on the current location of a user is of a similar interest as the retrieval of videos that provide some background information about nearby buildings, monuments, and landmarks. Moreover, a lot of research focuses on person-centric video annotation, such as human behavior and crowd analysis that are especially important for surveillance and assisted-living applications. As concluding remark, we like to state that there is an urgent need for holistic video annotation and retrieval systems that reach for an appropriate balance between functional richness and fine-grained metadata, that provide more added values for their users, and that are harmonized with existing systems and video portals.

In this thesis, we tried to work on techniques and infrastructures for such a holistic approach but obviously some points remain unsettled. As future work, we will investigate semi-local features in an integrated object recognition workflow during the TRECVID 2012 instance search task. In this process, we want to evaluate the impact of different segmentation approaches and the compatibility of semi-local features with best practice object recognition approaches. Furthermore, it might be interesting to apply semi-local features also for other computer vision tasks, for instance, in the area of robot navigation. It is planed to add further visual features and recognition approaches to the proposed recognition infrastructure and to extend the annotation systems by the additional use of audio. We intent to use the automatic approach selection for a broader range of multimedia content annotation and retrieval systems. In the context of automated object-of-interest annotation, we want to apply salient region detection as pre-processing step for the proposed annotation approach, and it is the final goal to integrate and test the approach in existing video platforms.

Bibliography

[1] ActionDetectionGroundTruth. *www.ims.tuwien.ac.at/sor/ActionDetectionGroundTruth.zip*.

[2] S. Agarwal and D. Roth. Learning a sparse representation for object detection. *ECCV*, pages 97–101, 2002.

[3] J.K. Aggarwal and Q. Cai. Human motion analysis: A review. In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 90–102. IEEE, 1997.

[4] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006.

[5] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.

[6] Alexia. *www.alexa.com*, 2012.

[7] T. Alisi, M. Bertini, G. D'Amico, A. Del Bimbo, A. Ferracani, F. Pernici, and G. Serra. Arneb: a rich internet application for ground truth annotation of videos. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 965–966. ACM, 2009.

[8] J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[9] AllMovie. *www.allmovie.com/*, 2012.

[10] M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 971–980. ACM, 2007.

[11] R. Arandjelovic and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 375–382. IEEE, 2011.

[12] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2294–2301. Ieee, 2009.

[13] F.G. Ashby and N.A. Perrin. Toward a unified theory of similarity and recognition. *Psychological review*, 95(1):124, 1988.

[14] J.A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 541–548. ACM, 2006.

[15] S. Ayache and G. Quénot. Video corpus annotation using active learning. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, pages 187–198. Springer-Verlag, 2008.

[16] B. Babenko, P. Dollár, and S. Belongie. Task specific local region matching. In *Computer Vision, 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[17] Baidu. *www.baidu.com/*, 2012.

[18] W. Bailer, P. Schallauer, H.B. Haraldsson, and H. Rehatschek. Optimized mean shift algorithm for color segmentation in image sequences. In *Proc. SPIE*, volume 5685, pages 522–529, 2005.

[19] Werner Bailer. A feature sequence kernel for video concept classification. In *Proceedings of 17th Multimedia Modeling Conference*, Taipei, TW, Jan. 2011.

[20] L. Ballan, M. Bertini, A. Bimbo, L. Seidenari, and G. Serra. Event detection and recognition for semantic annotation of video. *Multimedia Tools and Applications*, 51(1), 2011.

[21] L. Ballan, M. Bertini, A. Del Bimbo, and G. Serra. Enriching and localizing semantic tags in internet videos. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1541–1544. ACM, 2011.

[22] A. Baumberg. Reliable feature matching across widely separated views. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 774–781. IEEE, 2000.

[23] A. Baumberg. Reliable feature matching across widely separated views. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 774–781. IEEE, 2000.

[24] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417, 2006.

[25] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002.

[26] S. Bengio, F. Pereira, Y. Singer, and D. Strelow. Group sparse coding. *Advances in Neural Information Processing Systems*, 22:82–89, 2009.

[27] M. Bertini, A. Del Bimbo, A. Ferracani, L. Landucci, and D. Pezzatini. Interactive multi-user video retrieval systems. *Multimedia Tools and Applications*, pages 1–27, 2011.

[28] Bing. *www.bing.com*, 2012.

[29] S. Boll, P. Sandhaus, A. Scherp, and S. Thieme. Metaxa – context- and content-driven metadata enhancement for personal photo books. *Advances in multimedia modeling*, pages 332–343, 2006.

[30] R.M. Bolle, B.L. Yeo, and MM Yeung. Video query: Research directions. *IBM Journal of Research and Development*, 42(2):233–252, 1998.

[31] F. Bonin-Font, A. Ortiz, and G. Oliver. Visual navigation for mobile robots: a survey. *Journal of Intelligent & Robotic Systems*, 53(3):263–296, 2008.

[32] D. Bordwell, K. Thompson, and J. Ashton. *Film art: an introduction*, volume 7. McGraw-Hill New York, 1997.

[33] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM, 2007.

[34] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.

[35] F. Brémond, M. Thonnat, and M. Zúniga. Video-understanding framework for automatic behavior recognition. *Behavior research methods*, 38(3):416–426, 2006.

[36] D. Brezeale and D.J. Cook. Automatic video classification: A survey of the literature. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(3):416–430, 2008.

[37] G.J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[38] M. Brown and D.G. Lowe. Invariant features from interest point groups. In *British Machine Vision Conference, Cardiff, Wales*, pages 656–665, 2002.

[39] T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. In *Computer Vision and Pattern Recognition, 2011 IEEE Conference on*, pages 2225–2232. IEEE, 2011.

[40] M. Brut, S. Laborie, A.M. Manzat, and F. Sedes. A generic metadata framework for the indexation and the management of distributed multimedia contents. In *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, pages 1–5. IEEE, 2009.

[41] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[42] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[43] O. Carmichael and M. Hebert. Shape-based recognition of wiry objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(12):1537–1552, 2004.

[44] E.J. Carmona, J. Martínez-Cantos, and J. Mira. A new video segmentation method of moving objects based on blob-level knowledge. *Pattern Recognition Letters*, 29(3):272–285, 2008.

[45] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010.

[46] CAVIAR. *homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/*, 2012.

[47] M. Cha, H. Kwak, P. Rodriguez, Y.Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM, 2007.

[48] L.H. Chen, C.W. Su, C.F. Weng, and H.Y.M. Liao. Action scene detection with support vector machines. *Journal of Multimedia*, 4(4):248–253, 2009.

[49] X. Cheng, C. Dale, and J. Liu. Statistics and social network of youtube videos. In *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*, pages 229–238. IEEE, 2008.

[50] X. Cheng, K. Lai, D. Wang, and J. Liu. Ugc video sharing: Measurement and analysis. *Intelligent Multimedia Communication: Techniques and Applications*, pages 367–402, 2010.

[51] M. Christel and A. Hauptmann. The use and utility of high-level semantic features in video retrieval. *Image and video retrieval*, pages 588–588, 2005.

[52] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[53] P.I. Corke. The machine vision toolbox: a matlab toolbox for vision and vision-based control. *Robotics & Automation Magazine, IEEE*, 12(4):16–25, 2005.

[54] C. Cotsaces, N. Nikolaidis, and I. Pitas. Video shot detection and condensed representation. a review. *Signal Processing Magazine, IEEE*, 23(2):28–37, 2006.

[55] G. Csurka, C. Dance, F. Perronnin, and J. Willamowski. Generic visual categorization using weak geometry. *Toward Category-Level Object Recognition*, pages 207–224, 2006.

[56] G. Csurka and F. Perronnin. An efficient approach to semantic segmentation. *International journal of computer vision*, 95(2):198–212, 2011.

[57] S.J. Cunningham and D.M. Nichols. How people find videos. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 201–210. ACM, 2008.

[58] CVPR statistics. *www.cvpr2011.org/statistics*, 2012.

[59] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. Ieee, 2005.

[60] C. Damerval and S. Meignen. Blob detection with wavelet maxima lines. *Signal Processing Letters, IEEE*, 14(1):39–42, 2007.

[61] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and Y. Kompatsiaris. A survey of semantic image and video annotation tools. *Knowledge-driven multimedia information extraction and ontology evolution*, pages 196–239, 2011.

[62] R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

[63] Daum UCC. *www.daum.net/*, 2012.

[64] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296. ACM, 2010.

[65] R. de Charette and F. Nashashibi. Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 358–363. IEEE, 2009.

[66] F.D.M. de Souza, G.C. Chávez, EA do Valle, and A. de A Araujo. Violence detection in video using spatio-temporal features. In *Graphics, Patterns and Images (SIBGRAPI), 2010 23rd SIBGRAPI Conference on*, pages 224–230. IEEE, 2010.

[67] Delicious. *delicious.com*, 2012.

[68] J. Deng, A. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? *Computer Vision–ECCV 2010*, pages 71–84, 2010.

[69] J.J. DiCarlo and D.D. Cox. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341, 2007.

[70] S.K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros, and M. Hebert. An empirical study of context in object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1271–1278. IEEE, 2009.

[71] D. Djordjevic and E. Izquierdo. Kernels in structured multi-feature spaces for image retrieval. *Electronics Letters*, 42(15):856 – 857, 2006.

[72] D. Djordjevic and E. Izquierdo. Relevance feedback for image retrieval in structured multi-feature spaces. In *MobiMedia '06: Proceedings of the 2nd international conference on Mobile multimedia communications*, pages 1–5, New York, NY, USA, 2006.

[73] D. Doermann and D. Mihalcik. Tools and techniques for video performance evaluation. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 167–170. IEEE, 2000.

[74] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2005.

[75] J. Donahue and K. Grauman. Annotator rationales for visual recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1395–1402. IEEE, 2011.

[76] G. Dorkó and C. Schmid. Selection of scale-invariant parts for object class recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 634–639. IEEE, 2003.

[77] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. *ACM Transactions on the Web (TWEB)*, 1(2):7, 2007.

[78] A. Dyana and S. Das. Mst-css (multi-spectro-temporal curvature scale space), a novel spatio-temporal representation for content-based video retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(8):1080–1094, 2010.

[79] J.P. Eakins. Automatic image content retrieval-are we getting anywhere? In *ELVIRA-PROCEEDINGS*, pages 121–134, 1996.

[80] ECCV statistics. *www.ics.forth.gr/eccv2010/statistics.php*, 2012.

[81] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[82] H. Eidenberger. Kalman filtering for pose-invariant face recognition. In *Image Processing, 2006 IEEE International Conference on*, pages 2037–2040. IEEE, 2006.

[83] ETHZ Datasets. *www.vision.ee.ethz.ch/datasets/*, 2012.

[84] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy–automatic naming of characters in tv video. 2, 2006.

[85] M. Everingham, L Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[86] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009.

[87] J. Farquhar, S. Szedmak, H. Meng, and J. Shawe-Taylor. Improving bag-of-keypoints image categorisation: Generative models and pdf-kernels. *PASCAL Eprint Series*, 2005.

[88] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[89] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. *Computer Vision-ECCV 2004*, pages 40–54, 2004.

[90] V. Ferrari and A. Zisserman. Learning visual attributes. *NIPS*, Dec 2007.

[91] FFmpeg. *http://ffmpeg.org*, 2012.

[92] G.D. Finlayson. Color in perspective. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(10):1034–1038, 1996.

[93] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *IEEE Computer*, 28:23–32, 1995.

[94] Flickr. *www.flickr.com*, 2012.

[95] J. Flusser, T. Suk, B. Zitov, and Inc Ebrary. *Moments and moment invariants in pattern recognition*. Wiley Online Library, 2009.

[96] P.E. Forssen and D.G. Lowe. Shape descriptors for maximally stable extremal regions. In *Computer Vision, 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[97] D. Forsyth, J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. Finding pictures of objects in large collections of images. *Object Representation in Computer Vision II*, pages 335–360, 1996.

[98] J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[99] M. Frigo and S.G. Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

[100] FRVT. *www.nist.gov/itl/iad/ig/frvt-home.cfm*, 2012.

[101] B. Funt, K. Barnard, and L. Martin. Is machine colour constancy good enough? *Computer Vision-ECCV*, pages 445–459, 1998.

[102] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.

[103] S. Gammeter, A. Gassmann, L. Bossard, T. Quack, and L. Van Gool. Server-side object recognition and client-side object tracking for mobile augmented reality. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 1–8. IEEE, 2010.

[104] P. Geetha and V. Narayanan. A survey of content-based video retrieval. *Journal of Computer Science*, 4:474–486, 2008.

[105] G. Geisler and S. Burns. Tagging video: conventions and strategies of the youtube community. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 480–480. ACM, 2007.

[106] Y.L. Geng, D. Xu, J.Z. Yuan, and S.H. Feng. Two important action scenes detection based on probability neural networks. *Advances in Neural Networks-ISNN 2006*, pages 448–453, 2006.

[107] D.C. Gibbon, Z. Liu, and B. Shahraray. The miracle video search engine. In *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, volume 1, pages 277–281. IEEE, 2006.

[108] Goggles. *www.google.com/mobile/goggles/*, 2012.

[109] S.A. Golder and B.A. Huberman. Usage patterns of collaborative tagging systems. *Journal of information science*, 32(2):198–208, 2006.

[110] D.B. Goldman, B. Curless, D. Salesin, and S.M. Seitz. Interactive video object annotation. Technical report, Technical Report UW-CSE-2007-04-01, University of Washington, 2007.

[111] D.B. Goldman, C. Gonterman, B. Curless, D. Salesin, and S.M. Seitz. Video object annotation, navigation, and composition. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 3–12. ACM, 2008.

[112] I. Gordon and D. Lowe. What and where: 3d object recognition with accurate pose. *Toward category-level object recognition*, pages 67–82, 2006.

[113] D. Gossow, P. Decker, and D. Paulus. An evaluation of open source surf implementations. *RoboCup 2010: Robot Soccer World Cup XIV*, pages 169–179, 2011.

[114] V. Gouet and N. Boujemaa. Object-based queries using color points of interest. In *Content-Based Access of Image and Video Libraries, 2001.(CBAIVL 2001). IEEE Workshop on*, pages 30–36. IEEE, 2001.

[115] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1529–1536. IEEE, 2011.

[116] M. Grabner, H. Grabner, and H. Bischof. Fast approximated sift. *Computer Vision–ACCV 2006*, pages 918–927, 2006.

[117] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[118] Groupsac. *http://code.google.com/p/groupsac/*, 2012.

[119] L. Grunewaldt, K. Möller, and K. Morisse. Workflow for integrated object detection in collaborative video annotation environments. *Computational Science–ICCS 2006*, pages 565–572, 2006.

[120] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[121] B. Haasdonk and C. Bahlmann. Learning with distance substitution kernels. *Pattern Recognition*, pages 220–227, 2004.

[122] N. Haering, P.L. Venetianer, and A. Lipton. The evolution of video surveillance: an overview. *Machine Vision and Applications*, 19(5):279–290, 2008.

[123] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *Proceedings of the 16th international conference on World Wide Web*, pages 211–220, 2007.

[124] M.J. Halvey and M.T. Keane. Analysis of online video search and sharing. In *Proceedings of the eighteenth conference on Hypertext and hypermedia*, pages 217–226. ACM, 2007.

[125] A. Hampapur, R. Jain, and T.E. Weymouth. Production model based digital video segmentation. *Multimedia Tools and Applications*, 1(1):9–46, 1995.

[126] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[127] M. Hildebrand and J. van Ossenbruggen. Linking user generated video annotations to the web of data. *Advances in Multimedia Modeling*, pages 693–704, 2012.

[128] S. Hinz. Fast and subpixel precise blob detection and attribution. In *Image Processing, 2005. IEEE International Conference on*, volume 3. IEEE, 2005.

[129] C. Hofmann, U. Boettcher, and D.W. Fellner. Change awareness for collaborative video annotation. *Proceedings of the 9th International Conference on the Design of Cooperative Systems*, pages 101–118, 2010.

[130] D. Hoiem, A.N. Stein, A.A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *Computer Vision, 2007.IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[131] B. Hosack, C. Miller, and D. Ernst. Videoant: Extending video beyond content delivery through annotation. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pages 1654–1658, 2009.

[132] C. Hsu, C. Chang, and D. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, Taipei, 2003.

[133] M.K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[134] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(3):334–352, 2004.

[135] Y. Hu, D. Rajan, and L.T. Chia. Attention-from-motion: A factorization approach for detecting attention objects in motion. *Computer Vision and Image Understanding*, 113(3):319–331, 2009.

[136] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[137] Hulu. *www.hulu.com*, 2012.

[138] i-Lids. *www.homeoffice.gov.uk/science-research/hosdb/i-lids/*, 2012.

[139] ImageMagick. *www.imagemagick.org*, 2012.

[140] IMDB. *www.imdb.com/*, 2012.

[141] Intel Integrated Performance Primitives. *http://software.intel.com/en-us/articles/intel-ipp/*, 2012.

[142] JCS Jacques Junior, S.R. Musse, and C.R. Jung. Crowd analysis using computer vision techniques. *Signal Processing Magazine, IEEE*, 27(5):66–77, 2010.

[143] B. Jähne and H. Haussecker. *Computer vision and applications: a guide for students and practitioners*. Academic Press, 2000.

[144] M. Jahrer, M. Grabner, and H. Bischof. Learned local descriptors for recognition and matching. In *J. Pers, editor, Proc. of Computer Vision Winter Workshop CVWW08*, pages 39–46, 2008.

[145] M. Jain and CV Jawahar. Characteristic pattern discovery in videos. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pages 306–313. ACM, 2010.

[146] Y.G. Jiang, C.W. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 494–501. ACM, 2007.

[147] Joanneum Research. *www.joanneum.at/*, 2012.

[148] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[149] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *Computer Vision and Pattern Recognition, 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2. IEEE, 2004.

[150] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2. Ieee, 2004.

[151] D. Kelly. Methods for evaluating interactive information retrieval systems with users. *Found. Trends Inf. Retr.*, 3:1–224, January 2009.

[152] L.S. Kennedy, S.F. Chang, and I.V. Kozintsev. To search or to label?: predicting the performance of search-based automatic image classifiers. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 249–258. ACM, 2006.

[153] M. Kipp. Multimedia annotation, querying and analysis in anvil. *Multimedia Information Extraction*, 19, 2010.

[154] S.R. Klemmer, J. Li, J. Lin, and J.A. Landay. Papier-mache: toolkit support for tangible input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 399–406. ACM, 2004.

[155] T. Ko. A survey on behavior analysis in video surveillance for homeland security applications. In *Applied Imagery Pattern Recognition Workshop, 37th IEEE*, pages 1–8. IEEE, 2008.

[156] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A.C. Berg, and T.L. Berg. Baby talk: Understanding and generating simple image descriptions. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1601–1608. IEEE, 2011.

[157] A. Kumar and C. Sminchisescu. Support kernel machines for object recognition. In *Computer Vision, 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[158] N. Kumar, L. Zhang, and S. Nayar. What is a good nearest neighbors algorithm for finding similar patches in images? *Computer Vision–ECCV 2008*, pages 364–378, 2008.

[159] K. Lai and D. Wang. Towards understanding the external links of video sharing sites: measurement and analysis. In *Proceedings of the 20th international workshop on Network and operating systems support for digital audio and video*, pages 69–74. ACM, 2010.

[160] C.H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE, 2009.

[161] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[162] I. Laptev and P. Pérez. Retrieving actions in movies. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[163] Large Scale Visual Recognition Challange. *www.image-net.org/challenges/LSVRC/2012/index*, 2012.

[164] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization and segmentation. *Image and Vision Computing*, 27(5):523–534, 2009.

[165] Martha Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, and G.J.F. Jones. Automatic tagging and geotagging in video collections and communities. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pages 51:1–51:8, New York, NY, USA, 2011. ACM.

[166] Last.fm. *www.lastfm.de/*, 2012.

[167] G. Lavee, E. Rivlin, and M. Rudzsky. Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(5):489–504, 2009.

[168] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 371–378. ACM, 2007.

[169] S. Lazebnik, C. Schmid, and J. Ponce. A maximum entropy framework for part-based texture and object recognition. In *Computer Vision, 2005. Tenth IEEE International Conference on*, volume 1, pages 832–838. IEEE, 2005.

[170] S. Lazebnik, C Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. 2006.

[171] D Le, D. Zhu, S. Poullot, S. Satoh, V. Q. Lam, and D.A. Duong. National institute of informatics, japan at trecvid 2011. In *Proceedings of TRECVID Workshop*, Gaithersburg, MD, USA, 2011.

[172] Y.J. Lee and K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *Computer Vision and Pattern Recognition, 2011 IEEE Conference on*, pages 1721–1728. IEEE, 2011.

[173] Y.J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002. IEEE, 2011.

[174] B. Lehane, N.E. O'connor, and N. Murphy. Action sequence detection in motion pictures. In *The international Workshop on Multidisciplinary Image, Video, and Audio Retrieval and Mining*, 2004.

[175] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *Proc. BMVC*, 2006.

[176] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 775–781. IEEE, 2005.

[177] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *Computer Vision and Pattern Recognition, 2010 IEEE Conference on*, pages 1712–1719. IEEE, 2010.

[178] L.J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Computer Vision and Pattern Recognition, 2009. IEEE Conference on*, pages 2036–2043. IEEE, 2009.

[179] Y. Li, Y. Zhang, J. Lu, R. Lim, and J. Wang. Video analysis and trajectory based video annotation system. In *2010 Asia-Pacific Conference on Wearable Computing Systems*, pages 307–310. IEEE, 2010.

[180] Z. Li, R. Gu, and G. Xie. Measuring and enhancing the social connectivity of ugc video systems: a case study of youku. In *Quality of Service (IWQoS), 2011 IEEE 19th International Workshop on*, pages 1–9. IEEE, 2011.

[181] libVLC. *www.videolan.org/vlc/libvlc.html*, 2012.

[182] R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics*, 1(3):469–486, 2001.

[183] C.Y. Lin, B.L. Tseng, and J.R. Smith. Video collaborative annotation forum: Establishing ground-truth labels on large multimedia datasets. In *Proceedings of the TRECVID 2003 Workshop*, 2003.

[184] J. Lin and W. Wang. Weakly-supervised violence detection in movies with audio and video based co-training. *Advances in Multimedia Information Processing-PCM 2009*, pages 930–935, 2009.

[185] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.

[186] T. Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[187] A. Liu, J. Li, Y. Zhang, S. Tang, Y. Song, and Z. Yang. An innovative model of tempo and its application in action scene detection for movie analysis. In *Applications of Computer Vision, 2008. IEEE Workshop on*, pages 1–6. IEEE, 2008.

[188] D. Liu and T. Chen. Video retrieval based on object discovery. *Computer Vision and Image Understanding*, 113(3):397–404, 2009.

[189] H. Liu, D. Song, S. Rueger, R. Hu, and V. Uren. Comparing dissimilarity measures for content-based image retrieval. In *Aisa Information Retrieval Symp*, pages 44–50, 2008.

[190] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Computer Vision and Pattern Recognition, 2011 IEEE Conference on*, pages 3337–3344. IEEE, 2011.

[191] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[192] G. Loy and A. Zelinsky. Fast radial symmetry for detecting points of interest. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(8):959–973, 2003.

[193] W.L. Lu, J.A. Ting, K.P. Murphy, and J.J. Little. Identifying players in broadcast sports videos using conditional random fields. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3249–3256. IEEE, 2011.

[194] H. Luan, Y. Zheng, S.Y. Neo, Y. Zhang, S. Lin, and T.S. Chua. Adaptive multiple feedback strategies for interactive video search. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 457–464. ACM, 2008.

[195] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[196] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[197] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

[198] V. Malaxa and I. Douglas. A framework for metadata creation tools. *Interdisciplinary Journal of Knowledge and Learning Objects*, 1:151–162, 2005.

[199] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *Trans. on Circuits and Systems for Video Technology*, 11(6):703 –715, 2001.

[200] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[201] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40. ACM, 2006.

[202] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[203] J. Matas, D. Koubaroulis, and J. Kittler. Colour image retrieval and object recognition using the multimodal neighbourhood signature. *ECCV*, pages 48–64, 2000.

[204] Matlab: Computer Vision System Toolbox. *www.mathworks.com/products/computer-vision*, 2012.

[205] D. Maynes-Aminzade, T Winograd, and T. Igarashi. Eyepatch: prototyping camera-based interaction through examples. In *ACM Symposium on User Interface Software and Technology*, pages 33–42, 2007.

[206] MetaCrawler. *www.metacrawler.com*, 2012.

[207] F. Meyer. Color image segmentation. In *Image Processing and its Applications, 1992., International Conference on*, pages 303–306. IET, 1992.

[208] D. Mihalcik and D. Doermann. The design and implementation of viper. *University of Maryland - Institute for Advanced Computer Studies*, 2008.

[209] K. Mikolajczyk and J. Matas. Improving descriptors for fast tree matching by optimal linear projection. In *Computer Vision, 2007. IEEE 11th International Conference on*, pages 1–8. Ieee, 2007.

[210] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

[211] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *PAMI*, 2005.

[212] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, pages 43–72, 2005.

[213] D. Mitrović, M. Zeppelzauer, and C. Breiteneder. Features for content-based audio retrieval. *Advances in Computers*, 78:71–150, 2010.

[214] F. Moosmann, B. Triggs, F. Jurie, et al. Fast discriminative visual codebooks using randomized clustering forests. *Twentieth Annual Conference on Neural Information Processing Systems*, pages 985–992, 2006.

[215] J.M. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[216] N. Morsillo, G. Mann, and C. Pal. Youtube scale, large vocabulary video annotation. *Video Search and Mining*, pages 357–386, 2010.

[217] Y. Moses and S. Ullman. Limitations of non model-based recognition schemes. In *Proc. 2nd European Conf. on Computer Vision, Lecture Notes in Computer Science*, pages 820–828. Springer Verlag, 1992.

[218] X. Mu. Towards effective video annotation: An approach to automatically link notes with video content. *Computers & Education*, 55(4):1752–1763, 2010.

[219] M. Muja and D.G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, pages 331–340, 2009.

[220] M. Muja, R Rusu, G. Bradski, and D. Lowe. Rein - a fast, robust, scalable recognition infrastructure. In *International Conference on Robotics and Automation*, 2011.

[221] Multimedia Grand Challenge. *www.acmmm12.org/call-for-multimedia-grand-challenge-solutions/*, 2012.

[222] J. Mundy. Object recognition in the geometric era: A retrospective. *Toward Category-Level Object Recognition*, pages 3–28, 2006.

[223] J. Mutch and D.G. Lowe. Multiclass object recognition with sparse, localized features. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 11–18. IEEE, 2006.

[224] N. Naikal, A.Y. Yang, and S. Shankar Sastry. Informative feature selection for object recognition via sparse pca. In *Computer Vision, 2011 IEEE International Conference on*, pages 818–825. IEEE, 2011.

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

[226] A. Natsev. Multimodal search for effective video retrieval. *Image and Video Retrieval*, pages 525–528, 2006.

[227] Netflix. *www.netflix.com*, 2012.

[228] M.E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1447–1454. IEEE, 2006.

[229] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2161–2168. IEEE, 2006.

[230] A. Oerlemans and M.S. Lew. Retreivallab - a programming tool for content-based retrieval. In *International Conference on Multimedia Retrieval*, 2011.

[231] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[232] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527, 2007.

[233] X. Olivares, M. Ciaramita, and R. Van Zwol. Boosting image retrieval through aggregating search results based on visual annotations. In *Proceeding of the 16th ACM international conference on Multimedia*, pages 189–198. ACM, 2008.

[234] P. Over, G. Awad, J. Fiscus, B. Antonishek, M. Michel, A.F. Smeaton, W. Kraaij, and G. Quénot. Trecvid 2010 - an overview of the goals, tasks, data, evaluation mechanisms, and metrics. Technical report, National Institute of Standards and Tchnology, 2011.

[235] P. Over, G. Awad, J. Fiscus, B. Antonishek, M. Michel, A.F. Smeaton, W. Kraaij, and G. Quénot. Trecvid 2011–an overview of the goals, tasks, data, evaluation mechanisms, and metrics. Technical report, National Institute of Standards and Tchnology, 2012.

[236] C. Pantofaru, C. Schmid, and M. Hebert. Object recognition by integrating multiple image segmentations. *Computer Vision–ECCV 2008*, pages 481–494, 2008.

[237] R. Paredes, A. Ulges, and T. Breuel. Fast discriminative linear models for scalable video tagging. In *Machine Learning and Applications, 2009. ICMLA'09. International Conference on*, pages 571–576. IEEE, 2009.

[238] D. Parikh and C.L. Zitnick. The role of features, algorithms and data in visual recognition. In *Computer Vision and Pattern Recognition, 2010 IEEE Conference on*, pages 2328–2335. IEEE, 2010.

[239] M. Pedersoli, A. Vedaldi, and J. Gonzalez. A coarse-to-fine approach for fast deformable object detection. In *Computer Vision and Pattern Recognition, 2011 IEEE Conference on*, pages 1353–1360. IEEE, 2011.

[240] O. Pele and M. Werman. A linear time histogram metric for improved sift matching. *ECCV*, 2008.

[241] People Tracking Dataset. *www.ipf.kit.edu/downloads_People_Tracking.php*, 2012.

[242] F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1243–1256, 2008.

[243] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition, 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[244] J. Philbin and A. Zisserman. Object mining using a matching graph on very large image collections. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 738–745. IEEE, 2008.

[245] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face-recognition algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1090–1104, 2000.

[246] P.J. Phillips, W.T. Scruggs, A.J. O Toole, P.J. Flynn, K.W. Bowyer, C.L. Schott, and M. Sharpe. Frvt 2006 and ice 2006 large-scale results. *National Institute of Standards and Technology, NISTIR*, 7408, 2007.

[247] P.J. Phillips, H. Wechsler, J. Huang, and P.J. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[248] Photoshop. *http://www.photoshop.com/*, 2012.

[249] Picasa. *www.picasa.google.com*, 2012.

[250] PolyMeta. *www.polymeta.com/*, 2012.

[251] J. Ponce, T. Berg, M. Everingham, D. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. Russell, A. Torralba, et al. Dataset issues in object recognition. *Toward category-level object recognition*, pages 29–48, 2006.

[252] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[253] Qt. *http://qt.nokia.com/*, 2012.

[254] T. Quack, V. Ferrari, and L. Van Gool. Video mining with frequent itemset configurations. *Image and Video Retrieval*, pages 360–369, 2006.

[255] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 47–56. ACM, 2008.

[256] A. Rabinovich, A. Vedaldi, and Belongie A. Does image segmentation improve object categorization? Technical Report CS2007-090, U.C. San Diego, Computer Science and Engineering Department, 2007.

[257] W. Ren, S. Singh, M. Singh, and YS Zhu. State-of-the-art on spatio-temporal information-based video retrieval. *Pattern recognition*, 42(2):267–282, 2009.

[258] D. Renzel, Y. Cao, M. Lottko, and R. Klamma. Collaborative video annotation for multimedia sharing between experts and amateurs. In *Proceedings of the 11th International Workshop of the Multimedia Metadata Community*, pages 7–14, Barcelona, Spain, May 2010.

[259] C.J. Rijsbergen. *Information retrieval*. Butterworth, London, 1979.

[260] E. Rosten, R. Porter, and T. Drummond. Faster and better: a machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):105–119, 2010.

[261] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3):231–259, 2006.

[262] L.A. Rowe, J.S. Boreczky, and C.A. Eads. Indexes for user access to large video databases. In *Storage and Retrieval for Image and Video Database II, Proc. SPIE*, volume 2185, pages 150–161, 1994.

[263] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[264] B.C. Russell, W.T. Freeman, A.A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1605–1614. IEEE, 2006.

[265] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition? In *Computer Vision and Pattern Recognition, 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2. IEEE, 2004.

[266] B. Safadi, N. Derbas, A. Hamadi, F. Thollard, G. Quénot, H. Jégou, T. Gehrig, H.K. Ekenel, and R. Stifelhagen. Quaero at trecvid 2011: Semantic indexing and multimedi event detection (draft).

[267] A. Sanchez and V. David. Advanced support vector machines and kernel methods. *Neurocomputing*, 55(1-2):5–20, 2003.

[268] K. E. A. Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. volume 32, pages 1582–1596, 2010.

[269] F. Schaffalitzky and A. Zisserman. Automated scene matching in movies. *Image and Video Retrieval*, pages 186–197, 2002.

[270] B. Schiele and J.L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.

[271] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(5):530–535, 1997.

[272] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, pages 357–360. ACM, 2007.

[273] N. Sebe and M.S. Lew. Comparing salient point detectors. *Pattern recognition letters*, 24(1):89–96, 2003.

[274] G. Sela and M.D. Levine. Real-time attention for robotic vision. *Real-Time Imaging*, 3(3):173–194, 1997.

[275] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[276] Shazam. *shazam.softonic.com*, 2012.

[277] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[278] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[279] Shotdetect. *http://shotdetect.nonutc.fr/*, 2012.

[280] S. Siersdorfer, J. San Pedro, and M. Sanderson. Automatic video tagging using content redundancy. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 395–402. ACM, 2009.

[281] B. Sigurbjörnsson and R. Van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, pages 327–336. ACM, 2008.

[282] I. Simon, N. Snavely, and S.M. Seitz. Scene summarization for online image collections. In *Computer Vision, 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[283] J. Sivic and A. Zisserman. Efficient visual search for objects in videos. *Proceedings of the IEEE*, 96(4):548–566, 2008.

[284] A.F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006.

[285] J.R. Smith. The search for interoperability. *Multimedia, IEEE*, 15(3):84–87, 2008.

[286] C.G.M. Snoek and M. Worring. A state-of-the-art review on multimodal video indexing. In *Proceedings of the 8th Annual Conference of the Advanced School for Computing and Imaging*, volume 24, 2002.

[287] C.G.M. Snoek, M. Worring, O. de Rooij, K.E.A. van de Sande, R. Yan, and A.G. Hauptmann. Videolympics: Real-time evaluation of multimedia retrieval systems. *Multimedia, IEEE*, 15(1):86–91, 2008.

[288] D. Stavens and S. Thrun. Learning of invariant features using video. In *CVPR*, 2010.

[289] D.A. Steele. Method and system for video browsing on the world wide web, 1999.

[290] J. Stöttinger, A. Hanbury, N. Sebe, and T. Gevers. Do colour interest points improve image retrieval? In *Image Processing, 2007. IEEE International Conference on*, volume 1, pages I–169. IEEE, 2007.

[291] S.W. Suna, Y.C.F. Wanga, Y.L. Hunga, C.L. Changb, K.C. Chenb, S.S. Chenga, H.M. Wanga, and H.Y.M. Liaoa. Automatic annotation of web videos. In *Int. Conference on Multimedia and Expo (ICME),*, 2011.

[292] M.J. Swain and D.H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[293] S. Tabbone, L. Wendling, and J.P. Salmon. A new shape descriptor defined on the radon transform. *Computer Vision and Image Understanding*, 102(1):42–51, 2006.

[294] S. Tan, S. Chen, Z.-H. Zhou, and F. Zhang. Face recognition from a single image per person: A survey. *Pattern Recognition*, 39:1725–1745, 2006.

[295] C. Tanase and B. Merialdo. Efficient spatio-temporal edge descriptor. In *Advances in Multimedia Modeling, 18th International Conference*, volume 7131, page 210. Springer, 2011.

[296] M.J. Tarr and H.H. Bülthoff. Image-based object recognition in man, monkey and machine. *Cognition*, 67(1-2):1–20, 1998.

[297] C.H. Teh and R.T. Chin. On image analysis by the methods of moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(4):496–513, 1988.

[298] S. Tellex and D. Roy. Towards surveillance video search by natural language query. In *Proceeding of the ACM International Conference on Image and Video Retrieval*, page 38. ACM, 2009.

[299] S. Thaler, K. Siorpaes, E. Simperl, and C. Hofer. A survey on games for knowledge acquisition. Technical report, Semantic Technology Institute (STI), University of Innsbruck, Austria, 2011.

[300] TheMovieDB. *www.themoviedb.org/*, 2012.

[301] Torch-7. *http://www.torch.ch/*, 2012.

[302] Torch3vision. *http://torch3vision.idiap.ch/*, 2012.

[303] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997.

[304] A. Torralba, B.C. Russell, and J. Yeun. Labelme: Online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484, 2010.

[305] A. Toshev, B. Taskar, and K. Daniilidis. Object detection via boundary structure segmentation. In *Computer Vision and Pattern Recognition, 2010 IEEE Conference on*, pages 950–957. IEEE, 2010.

[306] K.N. Tran, I.A. Kakadiaris, and S.K. Shah. Part-based motion descriptor image for human action recognition. *Pattern Recognition*, 2012.

[307] M. Treiber. *An Introduction to Object Recognition: Selected Algorithms for a Wide Variety of Applications*. Springer-Verlag New York Inc, 2010.

[308] L. Trujillo and G. Olague. Automated design of image operators that detect interest points. *Evolutionary Computation*, 16(4):483–507, 2008.

[309] T. Tuytelaars. Dense interest points. In *Computer Vision and Pattern Recognition, 2010 IEEE Conference on*, pages 2281–2288. IEEE, 2010.

[310] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. In *Foundations and Trends in Computer Graphics and Vision*, volume 3, pages 177–280, 2008.

[311] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[312] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *british Machine vision conference*, pages 412–425, 2000.

[313] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. *Computer Vision–ECCV 2006*, pages 589–600, 2006.

[314] A. Ulges and T.M. Breuel. Can motion segmentation improve patch-based object recognition? In *Pattern Recognition, 2010 20th International Conference on*, pages 3041–3044. IEEE, 2010.

[315] A. Ulges, C. Schulze, D. Keysers, and T. Breuel. Identifying relevant frames in weakly labeled videos for training concept detectors. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 9–16. ACM, 2008.

[316] A. Ulges, C. Schulze, M. Koch, and T.M. Breuel. Learning automatic concept detectors from online video. *Computer vision and Image understanding*, 114(4):429–438, 2010.

[317] R. Unnikrishnan and M. Hebert. Measures of similarity. In *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 394–394. IEEE, 2005.

[318] J. Van De Weijer, T. Gevers, and A.D. Bagdanov. Boosting color saliency in image feature detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):150–156, 2006.

[319] J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders, and J.M. Geusebroek. Visual word ambiguity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1271–1283, 2010.

[320] V. Vapnik. *Statistical learning theory*. Wiley, New York, 1998.

[321] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.

[322] A. Vedaldi and B. Fulkerson. Vlfeat: an open and portable library of computer vision algorithms. In *ACM Multimedia*, 2008.

[323] R.C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 188–197. IEEE, 2001.

[324] R. Vezzani and R. Cucchiara. Video surveillance online repository (visor): an integrated framework. *Multimedia Tools and Applications*, 50(2):359–380, 2010.

[325] Video Annotation Prototype. *www.ims.tuwien.ac.at/ sor/VAP.zip*, 2012.

[326] VideoSurf. *www.videosurf.com/*, 2012.

[327] P. Viola and M. Jones. Robust real-time face detection. In *ICCV*, 2001.

[328] S. Wakamiya, D. Kitayama, and K. Sumiya. Scene extraction system for video clips using attached comment interval and pointing region. *Multimedia Tools and Applications*, 54(1):7–25, 2011.

[329] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, 2009.

[330] J.R. Wang and N. Parameswaran. Survey of sports video analysis: research issues and applications. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 87–90. Australian Computer Society, Inc., 2004.

[331] M. Wang, X.S. Hua, J. Tang, and R. Hong. Beyond distance measurement: constructing neighborhood similarity for video annotation. *Multimedia, IEEE Transactions on*, 11(3):465–476, 2009.

[332] M. Wang, G. Li, Y.-T. Zheng, and T.-S. Chua. Shottagger: Tag location for internet videos. In *Proc. of ACM ICMR*, 2011.

[333] X. Wang, X. Ma, and W.E.L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(3):539–555, 2009.

[334] X.J. Wang, L. Zhang, F. Jing, and W.Y. Ma. Annosearch: Image auto-annotation by search. *CVPR*, 2006.

[335] S.S. Wattamwar, S. Mishra, and H. Ghosh. Multimedia explorer: Content based multimedia exploration. *TENCON 2008-2008 IEEE Region 10 Conference*, pages 1–6, 2008.

[336] J. Weber and P. Lefevre, S.and Gancarski. Interactive video segmentation based on quasiflat zones. pages 265–270, 2011.

[337] J. Weber, S. Lefevre, and P. Gancarski. Video object mining: Issues and perspectives. In *Proc. IEEE Fourth Int Semantic Computing (ICSC) Conf*, pages 85–90, 2010.

[338] K. Weher and A. Poon. Marquee: A tool for real-time video logging. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pages 58–64. ACM, 1994.

[339] H. Wersing and E. Körner. Learning optimized features for hierarchical models of invariant object recognition. *Neural computation*, 15(7):1559–1588, 2003.

[340] Wikipedia. *www.wikipedia.org/*, 2012.

[341] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. *Computer Vision–ECCV 2008*, pages 650–663, 2008.

[342] S. Winder, G. Hua, and Brown M. Picking the best daisy. In *CVPR*, 2009.

[343] S.A.J. Winder and M. Brown. Learning local image descriptors. 2007.

[344] J. Winn and M. Everingham. The pascal visual object classes challenge 2007 annotation guidelines. *pascallin.ecs.soton.ac.uk/challenges /VOC/voc2007/guidelines.html*, 2007.

[345] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. Elan: a professional framework for multimodality research. In *Proceedings of LREC*, volume 2006, 2006.

[346] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition, 2011 IEEE Conference on*, pages 529–534. IEEE, 2011.

[347] G. Wong and H. Frei. Object recognition: the utopian method is dead; the time for combining simple methods has come. In *International Conference on Pattern Recognition*, 1992.

[348] WordNet. *http://wordnet.princeton.edu/*, 2012.

[349] J. Wu and J.M. Rehg. Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 630–637. IEEE, 2009.

[350] S. Wu, YF Li, and J. Zhang. Motion descriptor: A motion trajectory signature. In *Information and Automation, 2009. International Conference on*, pages 346–351. IEEE, 2009.

[351] X. Wu, W.-L. Zhao, and C.-W. Ngo. Towards google challenge: Combining contextual and social information for web video categorization. In *ACM Multimedia*, 2009.

[352] M. Wyl, H. Mohamed, E. Bruno, and S. Marchand-Maillet. A parallel cross-modal search engine over large-scale multimedia collections with interactive relevance feedback. In *ICMR*, 2011.

[353] L. Xie, A. Natsev, and J. Tesic. Dynamic multimodal fusion in video search. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1499–1502. IEEE, 2007.

[354] D. Yamamoto, T. Masuda, S. Ohira, and K. Nagao. Video scene annotation based on web social activities. *Multimedia, IEEE*, 15(3):22–32, 2008.

[355] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[356] S.L. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2249–2256. IEEE, 2010.

[357] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(4):13, 2006.

[358] YouKo. *www.youku.com*, 2012.

[359] YouTube. *www.youtube.com*, 2012.

[360] J. Yuan, H. Luan, D. Hou, H. Zhang, Y.T. Zheng, Z.J. Zha, and T.S. Chua. Video browser showdown by nus. *Advances in Multimedia Modeling*, pages 642–645, 2012.

[361] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1451–1458. IEEE, 2009.

[362] V. Zavřel, M. Batko, and P. Zezula. Visual video retrieval system using mpeg-7 descriptors. In *Proceedings of the Third International Conference on SImilarity Search and Applications*, pages 125–126. ACM, 2010.

[363] G. Zhai, G.C. Fox, M. Pierce, W. Wu, and H. Bulut. esports: collaborative and synchronous video annotation system in grid computing environment. In *Multimedia, Seventh IEEE International Symposium on*, pages 9–pp. IEEE, 2005.

[364] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical report, Microsoft Research, 2010.

[365] H. Zhang, J.E. Fritts, and S.A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.

[366] R. Zhang, R. Sarukkai, J.H. Chow, W. Dai, and Z. Zhang. Joint categorization of queries and clips for web-based video search. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 193–202. ACM, 2006.

[367] Z. Zhang, Y. Hu, S. Chan, and L.T. Chia. Motion context: A new representation for human action recognition. *Computer Vision–ECCV 2008*, pages 817–829, 2008.

[368] C.L. Zitnick and K. Ramnath. Edge foci interest points. In *Computer Vision, 2011 IEEE International Conference on*, pages 359–366. IEEE, 2011.