Volume 1

Media in Context

State of the Art in Cross-Media Analysis, Metadata Publishing, Querying and Recommendations

Responsible editor(s):
Patrick Aichroth, Johanna Björklund,
Florian Stegmaier, Thomas Kurz, Grant Miller

Volume 1

State of the Art in Cross-Media Analysis, Metadata Publishing, Querying and Recommendations

Patrick Aichroth, Johanna Björklund, Florian Stegmaier, Thomas Kurz, Grant Miller

About the project: MICO is a research project project partially funded by the European Commission 7th Framework Programme (grant agreement no: 610480). It aims to provide cross-media analysis solutions for online multimedia producers. MICO will develop models, standards and software tools to jointly analyse, query and retrieve information out of connected and related media objects (text, image, audio, video, office documents) to provide better information extraction results for more relevant search and information discovery.

Abstract: This Technical Report summarizes the state of the art in cross-media analysis, metadata publishing, querying and recommendations. It is a joint outcome of work packages WP2, WP3, WP4 and WP5, and serves as entry point and reference for technologies that are relevant to the MICO framework and the two MICO use cases.

Projekt Coordinator: John Pereira BA
Publisher: Salzburg Research Forschungsgesellschaft mbH, Salzburg, Austria
Editor of the series: Thomas Kurz | **Contact:** thomas.kurz@salzburgresearch.at
Issue: August, 2015 | **Grafik Design:** Daniela Gnad
ISBN 978-3-902448-43-9

Images are taken from the Zooniverse crowdsourcing project Plankton Portal that will apply MICO technology to better analyse the multimedia content. https://www.zooniverse.org

Online: A digital version of the handbook can be freely downloaded at http://www.mico-project.eu/technical-reports/

MICO (Media in Context) is a research project partially funded by the European Commission 7th Framework Programme (grant agreement no: 610480).

Responsible editors

Patrick Aichroth is working for Fraunhofer IDMT and focusing on user-centric, incentive-oriented solutions to the“copyright dilemma”, content distribution and media security. Since 2006, he is head of the media distribution and security research group. Within MICO, he is coordinating FHG activities, and is involved especially in requirements methodology and gathering, related media extractor planning, and system design and implementation aspects related to broker, media extraction and storage, and security.

Johanna Björklund is senior lecturer at the Department of Computing Science at Umeå University, and founder and COO of CodeMill, an IT-consultancy company specializing in media asset management (MAM). Her scientific work focuses on structured methods for text classification. In MICO she is leading the activities around a multi-lingual speech-to-text component.

Florian Stegmaier is a Postdoctoral Researcher at the Department for Distributed and Multimedia Information Systems (DIMIS) at the University of Passau. His research focuses on multimedia information retrieval and Semantic Web technologies. In Mico he is leading the work regarding multimedia annotations and metadata management.

Thomas Kurz is Researcher at the Knowledge and Media Technologies group of Salzburg Research. His research interests are Semantic Web technologies in combination with multimedia, human-computer interaction regarding to RDF metadata, and Semantic Search. In Mico he focuses on semantic multimedia retrieval and coordinates the overall scientific work.

Grant Miller is Social Media Manager for the Zooniverse crowdsourcing platform. As exoplanetary scientist he holds an PhD from the University of St Andrews. Grant is coordinating the work regarding cross media recommendation.

Responsible authors

Many different authors from all partners have contributed to this document. The individual authors in alphabetic order are:

- **Jakob Abeßer** (Fraunhofer Gesellschaft, FhG),
- **Patrick Aichroth** (FhG),
- **Suna Bensch** (Umeå University, UMU),
- **Emanuel Berndl** (University of Passau, UP),
- **Martin Berglund** (UMU),
- **Henrik Björklund** (UMU),
- **Johanna Björklund** (UMU),
- **Rafa Haro** (Zaizi Ltd., Zaizi),
- **Thomas Köllmer** (FhG),
- **Uwe Kühhirt** (FhG),
- **Thomas Kurz** (Salzburg Research Forschungsgesellschaft mbH, SRFG),
- **Alexander Loos** (FhG),
- **Grant Miller** (University of Oxford, UOX),
- **Ronny Paduschek** (FhG),
- **David Riccitelli** (InsideOut10, IO10),
- **Sebastian Schaffert** (SRFG),
- **Kai Schlegel** (UP),
- **Florian Stegmaier** (UP),
- **Andrea Volpini** (IO10),
- **Christian Weigel** (FhG),
- **and Rupert Westenthaler** (SRFG).

MICO – Volume

Volume 1

State of the Art in Cross-Media Analysis, Metadata Publishing, Querying and Recommendations

Patrick Aichroth, Johanna Björklund, Florian Stegmaier, Thomas Kurz, Grant Miller

ISBN 978-3-902448-43-9

Volume 2

Specifications and Models for Cross-Media Extraction, Metadata Publishing, Querying and Recommendations: Version I

Patrick Aichroth, Johanna Björklund, Kai Schlegel, Thomas Kurz, Grant Miller

ISBN 978-3-902448-44-6

Volume 3

Enabling Technology Modules: Version I

Patrick Aichroth, Johanna Björklund, Kai Schlegel, Thomas Kurz, Antonio Perez

ISBN 978-3-902448-45-3

Volume 4 – Publication date: December 2015

Specifications and Models for Cross-Media Extraction, Metadata Publishing, Querying and Recommendations: Version II

Patrick Aichroth, Johanna Björklund, Kai Schlegel, Thomas Kurz, Grant Miller

ISBN 978-3-902448-46-0

Volume 5 – Publication date: June 2016

Enabling Technology Modules: Version II

Patrick Aichroth, Johanna Björklund, Kai Schlegel, Thomas Kurz, Grant Miller

ISBN 978-3-902448-47-7

Contents

List of Figures

List of Tables

1 Introduction

This report represents the combined state of the art for work packages WP2, WP3, WP4, and WP5 of the MICO project. Its main objective is to provide the persons working on models and components in later phases of the project with a single entry point to lookup relevant conceptual background as well as concrete technologies as needed while implementing the generic MICO framework and particularly the two use cases. For this reason, the areas covered by this document have been carefully selected in accordance with the requirements gathering in work packages WP7 and WP8, and the system architecture in WP6. Any deliverables that are referenced within this report are free accessible on the web (`http://www.mico-project.eu/publications/`).

To summarize, the Zooniverse use case in WP7 will be mostly concerned with "object detection" and specifically "animal detection" in the *Snapshot Serengeti*[1] (SC-16), *Plankton Portal*[2] (SC-10) and *Galaxy Zoo*[3] (SC-14) projects projects, while the IO10 use case in WP8 will address quality assessment and annotation of commercial news video content (SC-02) and music content (SC-01). The reader is referred to MICO deliverables D7.1.1 and D8.1.1 for more details on the use cases. As the detailed requirements gathering is not fully completed, this report opens paths to different approaches in most areas, and is therefore very extensive. However, throughout the report there are many references pointing to the relevance to one or both of these use cases.

Since the report is a joint work of several work packages, we structured it mainly according to the tasks defined in the MICO Description of Work:

- Section 2 (Cross-Media Analysis) gives a thorough introduction into different approaches to analysing and extracting knowledge from media content and is the main outcome of work package WP2. In particular, it describes how analysis results are typically represented for further processing, it summarizes typical approaches to text analysis and extraction, and gives an overview over audio-visual analysis. Since the area is very broad and highly dependent on the actual domain, specific focus has been given to "animal detection" and A/V quality assessment, as these are the main topics of the two use cases.

- Section 3 (Metadata Publishing) is specifically concerned with the question how media content can be annotated, how analysis services can be described and how to represent trust and provenance in the context of analysis metadata and annotations. Since the MICO project is mostly concerned with Web content, the state of the art in this section focusses mainly on Semantic Web technologies to achieve these purposes. This section is the main outcome of work package WP3.

- Section 4 (Multimedia Querying) summarizes approaches for multimedia querying. It first gives an overview over different multimedia query languages that have been developed in the context of relational and semi-structured databases. In accordance with Section 3, it then describes the Semantic Web query language SPARQL and its extension mechanism, serving as the foundation for the MICO multimedia query language. This section is the main outcome of work package WP4.

- Section 5 (Multimedia Recommendations) investigates how to use the results from cross-media analysis, metadata publishing and multimedia querying for recommendations. It gives a summary

[1] `http://www.snapshotserengeti.org/`
[2] `http://www.planktonportal.org/`
[3] `http://www.galaxyzoo.org/`

of different approaches for recommender systems and specifically highlights those approaches that might be useful for the two use cases. This section is the main outcome of work package WP5.

- Section 6 (Implementations) finally gives a comprehensive list of concrete implementations that are available for many of the conceptual approaches described in the four main sections. It can serve as a reference for quickly looking up candidate technologies when needed.

Note that while these Sections follow mostly the respective work packages, the different areas have influenced each other significantly through the collaborative editing of the report. As most authors have also participated in the requirements analysis in work packages WP7 and WP8, the technologies and approaches described here should be of high relevance to the upcoming work in the MICO project.

2 Cross-Media Analysis

D2.1.1 Cross-media Analysis summarizes the state-of-the-art on cross-media extraction and identifies selected OSS and proprietary extractor implementations that can be used within the project.

In order to achieve this, the following topics are covered:

- High-level descriptions for all media content types, which are necessary to provide a baseline representation for all resources.
- Text Analysis, which covers all relevant topics related to the analysis of textual content
- Interactive Learning of Extractors, being especially relevant for textual extractors within the context of several Zooniverse showcases.
- Audio-visual Analysis, which covers all relevant topics related to the analysis of image, video and audio content.
- A brief conclusion, which includes the notion of the potential (and importance) of cross-modal approaches for improved accuracy and robustness, one of the key aspects within MICO.

The size and level of detail of the chapters on texual and audio-visual extratctors will thereby also reflect the priorities of the showcases described in D7.1.1 and D8.1.1: The most important showcases for Zooniverse, namely SC-16 (Snapshot Serengeti), SC-14 (Galaxy Zoo), and SC-10 (Plankton Portal) imply relevance especially for object and animal detection, species classification, with low-level feature extraction serving as a basis for several other visual analysis approaches. The two showcases for Inside-Out, SC-02 (News Videos) and SC-01 (Music), imply specific relevance for face recongition and music analysis. Texual analysis will be relevant across the board, as almost all showcases will provide relevant material, and the same goes for temporal video segmentation, and A/V error detection, which can serve as universal tools to segment, rank and filter audio-visual material for improved navigation and better user experience.

2.1 High-Level Representation of Media Content

Multimedia extraction is the process of analyzing and extracting information from multiple media resources such as text, audio, video, and images [May12a]. Analysis and information extraction from single media resources (e.g text, audio, image, video, social media) are well-studied fields and provide relatively successful tools and techniques. A history and state of the art of (multi-)media information extraction can be found in [May12b].

The breadth of possible types of resources makes it necessary to choose a flexible baseline representation. As logical reasoning across this representation is necessary it is appropriate to choose some discretized symbolic representation, but beyond that restrictions must be made only with great care. In the literature a variety of representations are used, strings and trees being common, but it stands to reason that when representing e.g. all relevant aspects of a two-dimensional image (e.g. neighborhood relations) in a string is necessarily difficult and artificial. As such state of the art systems often employ graphs (see e.g. the section on IRIS [HKH99, HKKZ95] below). Graphs not only allow for great modeling flexibility, but also include all relevant simpler representations as subclasses (i.e. strings and trees are graphs, and edge-less graphs permit simplistic sets of properties).

2.1.1 Graph Grammars for Complex Media Representation

Responsible partner / Author: UMU / Suna Bensch
Related Technology Enablers: TE-219

Grammars and their associated derivation trees have been used traditionally for syntactic and semantic analysis of natural language sentences or text. The syntactic and semantic structure of sentences is represented hierarchically in form of high-level syntactic and semantic structures. Recent approaches to text analysis with graph grammars can be found in [BDJvdM14, JAB+12]. Graphs are a convenient tool, not only for language analysis, but for any kind of application with a need to express structure and relations of arbitrary constituents. Furthermore, graph grammars allow for handling structural transformations. Thus, predestined applications for graph grammars are applications in which graphs are used as high-level data structures.

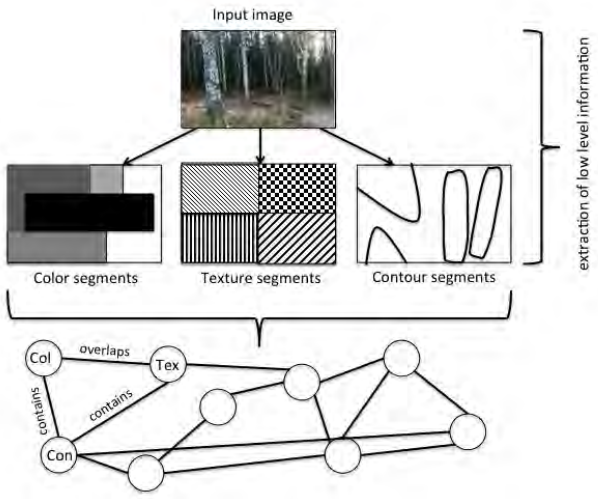
Many different types of graph grammars have been proposed, and several of them have been used in the context of media analysis. Stripping off any detail that distinguishes the formal definitions of different types of graph grammars, the principle is always the same: A graph grammar consists of rules that replace a left-hand side graph by a right-hand side graph. These rules are applied repeatedly in order to generate a graph in the graph language defined by the grammar. Graph grammars used in media analysis often allow to add attributes to nodes and edges of the graphs generated. Attributes can, e.g., carry spatial information or other information associated with the object represented by the node or edge in question.

We are interested in how and for which purpose, grammars and graph grammars (and their associated derivation trees and graphs) are used in text, image, and video analysis. In this section, we give a brief survey of works that use grammars or graph grammars for image and video analysis. In summary, it can be stated that, after low-level information extraction of image or video features, trees and graphs are used for syntactic and semantic high-level description of images and videos. Low-level features in images are, for example, texture and color; low-level features in video are, for example, location and velocity. High-level description of images and videos refer to, for example, composition of an image and event recognition in video sequences, respectively.

Prioritized MICO showcases that could benefit from image and video analysis techniques such as those surveyed here are the Zooniverse showcases and the InsideOut10 *News* showcase. For instance, in *Snapshot Serengeti* it could be used for the automatic detection of images with no classifiable animals in them, an animal species pre-classification into one of the 48 species of interest could be made, or the number of animals present in the image at hand could be estimated. This could be used to automatically free users from being presented images of no interest. In combination with text analysis, i.e., an analysis of user annotations, it could also be used to discover images that are worth a second look. In a similar way, these techniques can be used in Zooniverse *Plankton Portal*, *Galaxy Zoo*, and *Worm Watch*. In the InsideOut10 *News* showcase, video analysis techniques could be used for the automatic identification of, e.g., persons, products, logos, and brands.

Our long-term goal within the MICO project is to provide a formal model that combines different high-level descriptions of images, videos and text for the purpose of multimedia extraction (see technology enabler TE-408). We hope that such a model can decrease the semantic gap between human and machine interpretation of those multimedia resources.

Figure 1 First the image analysis phase extracts the low-level features color, texture and contour of segments of the input image. Then the spatial relations (called neighborhood relations) of the segments are computed and represented in a graph.



2.1.2 Systems Employing Graph Grammars

A previous effort in this direction is IRIS (**I**nformation **R**etrieval for **I**nformation **S**ystem), a system that automatically generates textual content descriptions of images for the purpose of image retrieval [HKH99, HKKZ95]. The provided textual content descriptions are annotations of the analyzed image itself and of objects occurring in the image. These image annotations facilitate the search for particular images (e.g. forest image, mountains etc.) in an image archive for a human user and can be used by any text retrieval system. The IRIS system combines techniques from computer vision and graph grammars for image analysis and object recognition in order to automatically annotate (objects in) the images. The system IRIS operates in two successive phases. The first phase performs image analysis extracting low-level information of the analyzed image and the second phase performs object recognition identifying primitive and more complex objects in the image. The image analysis phase extracts in three independent processes three low-level features of segments of the image, namely colour, texture and contour. The colour, texture and contour segments carry in addition to colour, texture and contour information also information about their position. Based on the information of these segments, topological relations (called neighborhood relations) of the segments are computed. Three neighborhood relations are distinguished, namely overlaps, meets and contains. The computed neighborhood relations of the segments are represented in a graph. Figure 1 illustrates the process of low-level information extraction and the graph representation of the neighborhood relations of the color, texture and contour segments. The object recognition phase identifies primitive and more complex objects using *neighbourhood-controlled node labelled and node attributed feature graph grammars* (1-NRCFGG) and a chart parser which is described in detail in [Kla94]. The primitive objects (e.g. forest, grass, sky) in the image are derived from the extracted low-level information and the more complex objects (e.g. forest scene) are derived from a combination of primitive objects.

The techniques of the system IRIS for image analysis are also used for content-based video analysis in [LMA99]. A complete video stream is divided into single shots and every shot is reduced to one representative still image. The still image is analyzed in the same way as described above. In addition, the system in [LMA99] employs superimposed text string extraction techniques as additional information about the video content. Moreover, the rules of the 1-NRCFGG are augmented with probabilities to deal with the several alternatives that are available during object recognition. The developed system in [LMA99] is for purpose of video retrieval from a video archive and supports and facilitates the archiving process, in particular, the content-based video annotation and retrieval.

An approach closely related to the IRIS approach was proposed in [ZLZ10], where by Zuzaňák et al. suggest to use attributed graph grammars for the recognition and description of image content. In this approach as well, ordinary image-processing techniques are used in a first phase to turn an image into a graph representation. Rules of an attributed graph grammar are then applied to this graph in order to extract high-level image information from this graph. The terminal nodes to which attributes are associated represent detected low-level features of the image (such as points, lines, and so forth) and the non-terminal symbols represent larger structures (for example, cross-walk). The authors illustrate their approach by showing how it can be used to detect cross-walks on images of street scenes.

In [HZ09] techniques from computer vision and *attribute graph grammars* (not to be confused with attributed graph grammars) are used to identify objects in images of man-made scenes such as building, offices, living spaces. Attribute grammars were introduced in [Knu68] to extract meaning from a parse tree, by transferring attribute values of terminal and non-terminal symbols, up and down the tree. In [HZ09] a corresponding notion of attribute graph grammars is used to identify rectangular objects like tables, floor tiles and windows in the given input image. Given an input image, an inference algorithm constructs a hierarchical parse graph showing the decomposition of the scene and the objects into components. The inference algorithm parses input images in the process of maximizing a Bayesian posterior probability and combines top-down and bottom-up hypotheses of possibly identified objects.

Driven by the need to make multimedia medical databases searchable by means of semantic criteria, the authors in [OTO05] use graph grammars to interpret complex X-ray images. The goal is to transform the visual information into semantic information that can be used for semantic indexing of important high-level objects visible in an image. To illustrate their approach, the authors discuss a so-called expansive graph grammar that describes both correct and pathological shapes and locations of wrist bones in X-ray images. Parsing an image according to the rules of the grammar yields an automatic assessment of the X-ray image's content, thus showing whether the wrist shown is normal or pathological.

A notion of spatial graph grammars allowing for efficient parsing algorithms is introduced in [KZZ06], extending the reserved graph grammars of Zhang et al. [ZZC01]. While these spacial graph grammars mainly seem to have been used for the specification of graphical user interfaces, they may turn out to be of general use for the analysis of scenes in which spatial relationships play a major role, owing to their efficient parsing algorithm.

The article [LGLW09] deals with representation and recognition of complex semantic events (e.g. illegal parking, stealing objects, etc.) in video sequences. The model in [LGLW09] is embedded in an intelligent visual surveillance system. Attribute graph grammars are used to semantically analyze activities in a car park. The grammar is used to represent all possible variations and hierarchical

composition of an event. For example, the semantic event "a coming car is picking up a man" is decomposed into so-called event components with spatio-temporal constraints "waiting" and "picking up" and "moving away". The "picking up" event is further decomposed into so-called event primitives "approach" and "enter" with the temporal constraint that this events have to happen in sequential order. These event primitives are further decomposed into atomic event primitives (e.g. stop, move, stay) which are learned from an object-trajectory table describing mobile object attributes (by low-level features such as location, velocity, and visibility) in a video sequence. The attributed graph grammar and the detected objects represent a set of possible interpretations of an activity. A probability distribution over this set is expressed as a Markov random field.

In [DH12, DH09] a framework for detecting activities from video sequences for the purpose of automatic surveillance is developed. Automatic surveillance involves recognizing multiple activities that are possibly interleaved. The activities and their (mutual) constraints are used to provide a formal framework that helps interpreting activities in a given noisy visual input. The authors in [DH12, DH09] use *attribute multiset grammars* (AMGs) in order to represent the possible interpretations of a given activity (e.g. the activity in a bicycle rack). Given an input video, detectors retrieve a multiset D of detections (e.g. person, object, bicycle, etc.) with the help of computer vision techniques. These detections are parsed according to an AMG, which defines the hierarchical composition of the observed hierarchy. Each parse tree has a posterior probability in a Bayesian sense. Finding the best parse tree, or in other words, the best explanation of an activity, corresponds to finding the maximum a posteriori labeling of the Bayesian network. The framework is tested for two applications: the activity in a bicycle rack and around a building entrance.

2.1.3 Advanced Processing on Restricted Representations

While the flexibility of the graph representation and the expressiveness grammatical systems working on graphs, as presented in e.g. [BDJvdM14], are indisputable, the power comes at a cost. A variety of baseline tasks, such as emptiness checking and parsing are in general cases extremely difficult [BBD10]. As such, while graphs are central in the representation for the purposes of generality, and graph grammars are a great tool for very expressive modeling on them, most practical extraction tasks must by necessity operate on a simpler level. The most immediate example is text analysis. In a cross-media context, natural language text will play a part in most use cases, in at least three ways

1. as machine-readable, though natural language, text (e.g. Zooniverse posts),

2. as spoken text, which through speech to text transcription is transformed into machine-readable form (see technology enabler TE-403),

3. as written, but not machine-readable text, with optical character recognition as a relevant but not currently considered technology enabler extension.

Importantly item 2 will benefit from a graph representation in that a transcription produces a weighted directed acyclic graph to represent different possible interpretations. However, state-of-the-art natural language processing is often an expensive process aimed at already machine-readable strings, meaning that the graph is treated as a packed representation from which a few candidates are extracted rather than viewed as a canonical representation in its own right. A deeper look into the representations and techniques in text analysis is given in Section 2.2.

Another important aspect is the interactive learning of extractors. Here the grammatical sophistication needs to be limited to make the task tractable. Operating on the level of graph grammars is too difficult, as even the class of context-free languages is widely considered too strong for practical grammatical inference [DLH05]. Here the grammatical restrictions leave mostly finite state approaches practical, which are well suited for tree representations, providing powerful tools for learning queries on hierarchical structures, see Section 2.3 for more on the representations and state of the art techniques in this area.

2.2 Text Analysis

Responsible partner / Author: UMU / Suna Bensch, Martin Berglund, and Johanna Björklund
Related Technology Enablers: TE-212, TE-213, TE-215, TE-216, TE-217, TE-220, TE-221

This subsection outlines the textual analysis aspect of cross-media analysis. The analysis of text presents some distinctive challenges, where audio, images and video present difficulty in extracting any symbolically valid information at all, text is more directly accessible. That is, with special consideration for the case where text is extracted from audio using speech-to-text systems, or from images using optical character recognition, the symbolic surface contents of text is immediately available. In addition, from the graph perspective the raw text is structurally simplistic, in that a certain string can be represented as a single chain graph (or the set of possible strings as the path language of the graph), but the deeper meaning of the text may be very complex. The problem is that text will often encode complex ideas, where an image of a real-world object seldom represents anything except the object, the meaning of text can be deep and varied. For this reason different, and more specific, techniques are required.

Text analysis spans a range of tasks, including document categorization, authorship attribution, readability assessment, sentiment and discourse analysis, language identification, and named entity recognition [JM00], each of which has its own set of tools and techniques (see table below). Two paradigms can be discerned, namely, based on symbolic and stochastic computing. The former paradigm investigates natural languages using formal language theory and generative grammar with its origins dating back to Chomsky [Cho56], as well as techniques from artificial intelligence. The latter relies on statistical models such as Bayesian networks and was pioneered by Harris and others [Har62]. Over time, these approaches have come together in the sense that the data-driven models are given more structure, at the same time as the automata-based models and generative grammars are enhanced with weights and probabilities. Examples of such hybrid approaches for authorship attribution include the work by [LZ09] and [RKM10]. Lin and Zhang use machine learning to infer stochastic grammars to distinguish between authors, whereas Raghavan et al. train probabilistic context-free grammars. This use of machine learning is today quite widespread thanks to high-performance computing and the availability of large annotated data sets such as the Penn Treebank [MSM93] and the Prague Dependency Treebank [Haj98].

2.2.1 Sentiment Analysis

In the Zooniverse Snapshot Serengeti and Galaxy Zoo showcases, one of the goals is to decide when to call the attention of the scientists to a post or series of posts. For this purpose, it can be useful to decide if there is a strong polarity to the discussion, e.g. a heated debate, in other words, to do a sentiment analysis on the text. The corresponding tools will be implemented as technology enabler TE-402.

Figure 2 Common tools and techniques for different NLP tasks.

Task	Common techniques	Software tools
Document categorization	SVM, Naive Bayes, Kernel methods	Intellexer Categorizer, NetOwl DocMatcher
Authorship attribution	SVM, Naive Bayes, Weighted automata	Signature, JGAAP
Readability assessment	Rule-based systems, Lexical analysis	the Readability Test Tool, Microsoft Word's built in statistics
Sentiment analysis	SVM, Naive Bayes, Kernel methods	SAS Sentiment Analysis, Lexalytics, Stanford NLP Sentiment
Discourse analysis	Formal grammars, Decision trees, Textual entailment, Ontologies	NVivio, DATool
Language identification	Lexical analysis, n-gram models	Rosette Language Identifier, Virtual Salem, Cybozu Language Detection
Named entity recognition	Regexps, Markov models	Stanford NER, LingPipe, OpenNLP, Freeling

Sentiment analysis [DC01, Ton01] consists in establishing the writer's judgement, affective state, or intended emotional communication. The simplest form is perhaps polarity-detection, in which we are satisfied to separate between positive and negative statements. In more advanced forms, the scale may be broken down according to some affect theory, into classes such as happy, sad, angry, and so forth. We can also try to detect the subject who has the opinion, and the object that the opinion is about. Common applications of sentiment analysis include detection of antagonistic language in online communication [Spe97], advert placement [JLMT07], discarding subjective opinions in information extraction [RWP05], automatic question answering [LSHN05], and summarization of reviews [PL08]. Within the scope of MICO, sentiment analysis can be useful for calling the moderators attention to scientific discussions, detecting arguments, and for generating features, e.g. image classifiers.

Many sentiment analysis systems are based on a Bag-of-Words (BoW) approach: Each sentence is seen as a set of words, the order of which is disregarded, and the polarity of the sentence depends on how well its words fit to pre-defined lists of positive and negative keywords. A difficulty here is that polarized opinions can be expressed almost as well without charged words. Take for example the quote "She runs the gamut of emotions from A to B" (Dorothy Parker about Kathrine Hepburn), this is clearly not a praise, although non of the words in the sentence is inherently negative [PL08]. Another obstacle is sarcasm, in which the speaker says the opposite of what he or she is actually thinking. Davidov et al try to detect sarcasm by looking for strongly positive statements in a negative context, or vice-versa [DTR10]. As can be expected, the sentiment analysis is context sensitive: "read the book" is a positive thing to see in a book review, but a negative thing in a movie review [PL08].

Researchers at Stanford University have published a corpus, the *Stanford Sentiment Treebank*, with sentiment labels for some 215 000 phrases in the parse trees of approx. 12 000 sentences. The corpus is based on a dataset of movie reviews collected and published by Pang and Lee [PL08], and later parsed with the Stanford Parser [KM03]. The same team of researchers introduce the *recursive neural tensor network*, a type of neural network in which a tensor-based composition functions are used at all nodes. By training this model on the sentiment treebank, the authors obtain a softmax classifier

Figure 3 Question (a) is declarative, question (b) imperative, while (c) and (d) are more or less rhetorical.

(a)	I was wondering how you sign up
(b)	Tell me where you see it
(c)	What the chances of both of my phones going off at the same time?
(d)	How has everyone's day gone so far? Today is going too fast for me!

with an accuracy of 85.4% for single-sentence sentiment-polarity classification. This improves on the state of the art by more then five percent.

2.2.2 Question Detection

As mentioned previously, one of the requirements from Zooniverse is the ability to decide when to call the researchers' attention to a discussion. One obvious trigger is unanswered questions, which leads us to the field of automatic question detection (to be covered by technology enabler TE-401). An obvious first step is to take sentences ending with question marks to be questions. This yields a high precision of over 97% [CWL$^+$08], but misses declarative questions (Table 3 a) and imperative questions (b). According to Kwong and Yorke-Smith, approximately one out of five questions are of these kinds in the Enron corpus of email conversations [KYS09]. It also means that rhetorical questions are taken to literally (c), and questions that are primarily if not exclusively meant to introduce a new topic (d) [DP11].

Margolis and Ostendorf [MO11] compare methods for detecting questions in spoken conversation. According to their results, models trained on unlabelled internet dialogues reach 90% of the performance of models trained on labelled, domain-adapted text. The difference was greatest for declarative questions, but since we will focus on text, where punctuation symbols are available, we expect the gap to be smaller. The authors considered as questions all complete utterances labeled with one of the labels *wh*, *yes-no*, *open-ended*, *or*, *or-after-yes-no*, or *rhetorical question* (see Table 4).

Figure 4 Example questions given by Margolis and Ostendorf [MO11] for the categories proposed by Shriberg et al. [SDB$^+$04]

yes-no	did you do that?
declarative	you're not going to be around this afternoon?
wh	what do you mean um reference frames?
tag	you know?
rhetorical	why why don't we do that?
open-ended	do we have anything else to say about transcription?
or	did they use sigmoid or a softmax type thing?
or-after-YN	or should i collect it all?

.

Boakye et al. [BFHT09] also investigate question detection in spoken dialog, using a features from the classes lexico-syntactic (words, part-of-speech tags, and syntactical fragments), turn-based (utterance length and speaker changes), and pitch-related (F0 statistics and slope). They take particular

interest in features stemming from parsed versions of the utterances, as this according to them is a novel aspect. The authors come to the conclusion that the lexio-syntactic features contribute the most, in particular word n-grams followed by syntactical features, and that turn-based and pitch-related features are secondary sources of information. Wang and Chua confirm the usefulness of syntactical features in question detection. In particular, order in which the noun phrase and the verb phrase appear is helpful for telling questions and non-questions apart [WC10].

2.2.3 Automatic Moderation

Veloso et al. [VMM+07] consider automatic moderation of online forums with associative classification, and include of social-network features in the feature sets. In their experiments, they use a corpus of manually annotated comments collected from the Slashdot forum[4]during a period of two months. The corpus contains some 300 000 entries written by approx. 42 000 distinct users. The authors reach the conclusion that a lazy classifier that induces a new model for every input comment achieves better accuracy than an eager version, that classifies all comments with a single model.

The *salient* parts of a document are those that the reader is likely to find particularly moving or provocative [Del09]. Relevance is one important factor that decides salience, but it is neither necessary nor sufficient. Misspellings, imbalanced arguments, and ambiguity also matters, and the notion is strongly contextual. Delort exploit user comments to detect salient parts. The idea is that when the user comments on a text, he or she will focus on the salient parts, and use related wording. By analysing the comments, it should be possible to identify some parts of the text as more salient than others. The author obtained an fscore of 0.65% on a corpus of blog post and associated comments.

2.2.4 Automatic Summarization

Automatic summarization consists in reducing the length of a text, while preserving its semantics. A comprehensive overview is given in [NM11], which covers much of the field's 50-year history. Our interest in summarization is motivated by InsideOut10's usecase on News, in which it would be helpful to provide short descriptions of news items. Summarization could also be useful for abbreviating lengthy forum discussions in the Zooniverse usecases.

There are two general approaches, namely *extraction* and *abstraction*. The former method consists in selecting informative passages from the text and have these represent the text as a whole. The latter method is more complex, in that it builds a semantic representation of the textual content and then generates a summary, thereby paraphrasing the source document. At present, most practical summarization systems use extractive summarization.

The extraction of key phrases can be seen as a supervised machine learning problem. The user labels training sentences depending on how well they capture the meaning of the document, and the machine tries to find a pattern. It may, for instance, discover that headings are likely to be keyphrases, and that so are the first and last sentences of a paragraph. Typically, a threshold parameter is used to regulate how long a summarization is generated.

Extractive and abstractive summarization approaches can be applied to single documents or multiple documents. Single document summarization methods deliver a single summary of one document

[4] `www.slashdot.org`

(i.e. news item, scientific article), whereas multi-document summarization provide a single summary of several documents. Multi-document summarization has gained interest since the mid 1990s due to the use cases on the web, in particular for summarization of news articles [NM11, DM07]. Major difficulties in multi-document summarization are overlapping, supplementing and contradicting information sources and main research questions address handling redundancy, recognizing novelty across documents, and providing coherent and complete summaries [DM07]. The authors in [KS12] give an overview of four notable extractive methods to multi-document summarization, namely feature-based, cluster-based, graph-based, and knowledge-based.

The feature-based method identifies the most relevant sentences to form a summary. The relevance of a sentence is determined by features such as word frequency, sentence location, sentence length, cue words, and proper nouns. The feature based method is popular due to its simplicity and straightforwardness. However, the method fails to detect important and relevant information across documents and does not incorporate contextual information.

The cluster-based method groups similar sentences to clusters. The most common technique to determine similarity between a pair of sentences is the cosine similarity measure, where sentences are represented as a weighted vector of term frequency-inverse document frequency (short tf-idf). The sentence clustering is followed by the sentence selection, where sentences are selected from each cluster based on the closeness to the top ranking tf-idf. The selected sentences form the summary. Notable web-based news clustering systems are Google News[5] and Columbia Newsblaster[6] [DM07]. Cluster-based methods handle redundancy across documents relatively successfully, but during the clustering process sentences are ranked according to the similarity to the cluster centroid, which simply represents frequently occurring terms. Thus, cluster-based methods cannot incorporate contextual information either.

The graph-based method represents sentences as vertices in a graph and the similarities between sentences are represented by weighted edges. The similarity between sentences across documents is most often determined with the cosine measure with some predefined threshold parameter. Two popular graph-based ranking algorithms are the HITS algorithm and the Google's PageRank, which both are used for web-link analysis and social network studies.

The knowledge-based method is applicable to specific domains. Most documents are about a particular topic or event and these belong to a particular domain with common background knowledge (i.e. ontology). Ontologies are able to capture hidden semantic information. On the other hand, the knowledge-based method is domain-specific and requires input from domain experts.

There appears to be no advanced open source toolkits dedicated to automatic summarization, but the Python Natural Language Toolkit provides much of the required functionality.

2.2.5 Named Entity Extraction

Named Entity Extraction (NEE) is the task of identifying parts of text as referring to specific objects, or entities, by some identification. Notably, in the preceding sentence, a good NEE tool would automatically realize that the initial portion "Named Entity Extraction (NEE)" does not in fact serve

[5] https://news.google.com/
[6] http://newsblaster.cs.columbia.edu/

the role as grammatical text, but is rather an atomic reference to some entity, in this case a research field.

The research field is fairly young, having only been named in 1996 (though being a natural problem it was of course considered in isolated systems and papers before this point). In many simple cases simple heuristics are sufficient, such as noting the additional capitalization as well as parenthesized acronym in "Named Entity Extraction (NEE)", but state of the art techniques are primarily focused on supervised learning, using standard techniques such as hidden Markov models, support vector machines and conditional random fields. See [NS07] for a survey on both the history of the field and a thorough overview of the techniques.

In the context of MICO, named entity recognition presents both opportunities and challenges. Trained extractors, such as the Stanford Named Entity Recognizer (see `http://nlp.stanford.edu/software/CRF-NER.shtml`) are available. The Stanford Named Entity Recognizer is based on training conditional random fields, see [SM12] for more information, and is both straightforward to integrate and comes with pre-trained models that can be immediately applied. In some cases, however, the named entities selected for training will be limited, the Stanford Named Entity Recognizer has models for the so called ENAMEX class, which recognizes names of people, organizations and locations, up to a 7-class model recognizing in addition times, money, percentages and dates. As can be seen by this a certain rigidity in focus exists, and for example, the Zooniverse use-cases a very obvious named entity would be names of animals. This may require additional training.

This section would be amiss not to mention that some work on named entity extraction has been done specifically for cross-media setting, such as in [BCD05]. However, in the context of MICO it is likely sufficient to assume that text extraction has already been performed in a different component.

2.2.6 Ontologies and data models for NLP

NIF (Nlp Interchange Format) [HLAB13] is an RDF based format that aims to achieve interoperability between Natural Language Processing (NLP) tools, language resources and annotations. To formally represent NLP annotations NIF first defines an URI scheme based on RFC 5147[7] that allows to reference Strings (words, phrases, sentences ...) as RDF resources[8] and second it defines/uses a set OWL ontology that allows to formally describe such Strings. The String Ontology is the core vocabulary to describe a String as part of a document and possible a sub-String of an other one. Structured Sentence Ontology (SSO) allows to define Strings as Sentences, Phrases or Words and its relationships. The Ontologies of Linguistic Annotation (OLiA) [CS14] defines a Reference Model with classes for linguistic categories (e.g. Noun, Determiner ...). Multiple Annotation Models formalize annotation schemes and tag sets used by different corpora and languages.

Finally for every Annotation Model, there is also a Linking Model that aligns it with the OLiA Reference Model. This allows to use the OLiA Reference Model to access/query NLP annotations using different annotation models (e.g. NLP annotations generated by different NLP tools/framework). It also supports queries for Words with a specific lexical category (e.g. ProperNouns) in texts with different languages. NIF uses OLiA for representing the POS annotations of words. NERD (Named Entity Recognition and Disambiguation) [RT12b, RTHB12] defines both a data model for describing

[7] `http://tools.ietf.org/html/rfc5147`

[8] For example, the String "Semantic Web" occurring in the file `example.txt` can be referenced by the call `http://www.example.org/example.txt#char=45,57`

detected Named Entities as well as an ontology for the types of Named Entities. This hierarchy of Named Entity types is also aligned with several other ontologies and common datasets. NIF uses NER to link Strings with entities as well as with the NER type.

2.3 Interactive Learning of Extractors

Responsible partner / Author: UMU / Henrik Björklund
Related Technology Enablers: TE-212, TE-213, TE-215, TE-216, TE-217, TE-218

In order to systematically extract certain information from, e.g., web pages, we need *queries* that define what information is to be extracted and *query processors* that do the actual extraction. What information can be extracted depends on what formalism we use for expressing queries.

Independent of formalism, one possibility for defining queries is to have human beings write them. In many situations, this works quite well, but there are severe drawbacks to this approach. It requires specialized skills from the people constructing the queries and is often time-consuming and expensive. The process is error-prone and most often it has to be repeated when, e.g., a web page design changes.

For these reasons, it is desirable to employ *machine learning* in defining queries. Learning, however, throws up another set of difficulties. In the first place, learnt queries are, for the most part, of lower quality than those defined by humans. Secondly, most machine learning techniques require huge amounts of training data in order to achieve even reasonable results.

In a setting where, e.g., a system administrator for a website wants to define queries that extract information relevant to the venture, data will normally be relatively sparse. In particular, there will be no more annotated data than what the administrator has time to annotate herself. This means that relying on pure machine learning will yield very poor results.

Interactive learning When annotated data is sparse, interactive learning can be a viable alternative. The main idea is to use the knowledge of a domain expert, e.g., the system administrator mentioned above, as efficiently as possible, since human time is normally a very expensive resource. The system expert annotates some data, but not necessarily very much. The annotated data is used to learn a query. The expert then views the results of running the query on a test data set and gives additional input to the learning system. This process is iterated until the expert is satisfied with the query result.

The concept of iterative learning has been studied extensively by, e.g., Small [Sma09]. This thesis looks at how interactive learning can be applied to a number of different learning tasks, focusing on NLP applications. It concludes that interactive learning can substantially assist a domain expert in encoding world knowledge into the learning process. Importantly, Small writes that "interactive encoding of modeling information through feature engineering often leads to better performance than simply acquiring additional labeled data." He demonstrates the effectiveness of interactive learning on a semantic role labeling and an information extraction task.

Tree-shaped data Sidestepping the issue of graphs, most of the textual data available on the web is structured hierarchically, i.e., as HTML, XML or Json documents. Such hierarchical structures be abstractly represented by trees. As seen in Section 2.1, data that has been extracted from images, video,

etc., while sometimes represented as graphs, are often represented as trees (or potentially have relevant spanning trees). The same is true of syntactic and semantic information that has been extracted from pure text, most notably the syntactic information extracted by natural language parsers. Whether phrase structure parsers or dependency parsers are used, the resulting structures are tree- or graph shaped, see, e.g., [KM03, dMMM06, KMN09].

For these reasons, learning grammars, automata, and transducers for trees and graphs are of particular interest. There is already an extensive literature on machine learning for regular tree languages, see, e.g., [DH03, Dre09] and, to a lesser extent, for graph grammars [KHC07, KHC08]. Much less has been written, on the topic of interactive learning for these structures but there are some notable exceptions, primarily from the field of web wrapper induction.

Web wrappers A *web wrapper* is a program that gathers information from web pages (or possibly web based XML repositories, etc.) and transforms it into some form structured for data analysis, commonly XML or relational. The need for web wrappers has arisen since large amounts of data, e.g., about the weather, travel time tables, stock prices, and sports results, are made available on web pages, but embedded into other content and formatted for human viewing. Someone who wants to keep track data from a certain set of web pages can write a web wrapper that continually visits the web pages, extracts the relevant information, and stores it in the users own database.

As discussed above, it is expensive and time-consuming to have human experts writing web wrappers. This, combined with a growing demand, has sparked research and development of systems for assisting wrapper design.

One such system is Lixto, first presented in an article by Baumgartner, Flesca, and Gottlob in 2001 [BFG01]. It lets the user mark elements on a web page in a graphical user interface. The user choices are represented by *filters* that select similar elements. The user can also impose *conditions* on when filters can be applied. In this way, each added filter increases the amount of information selected, while each condition limits the information selected by some filter. Together, the filters and conditions are translated into a datalog-like language called Elog, which is used internally in the system. The user can also specify an XML format to store the extracted data in.

We note, without citing all the relevant articles, that the Lixto system has undergone substantial development and more functionality has been added since the original version was presented and an information extraction company has been built around it.[9]

Even if Lixto is based on graphical interaction, it is not obvious that it is easy enough to use for an end-user with limited technical knowledge [GMTT06]. Re-usability is also limited by the fact that the system is now commercial.

Node-selecting tree transducers In an article on *Interactive learning of node selecting tree transducers*, Carme et al. also tackle the problem of learning *web wrappers* [CGLN07]. Their approach is to limit the power of the query language. In this setting, a *node-selecting tree transducer* is a deterministic finite unranked tree automaton except that it marks all nodes that have been visited in an accepting state. The subtrees of such nodes are then selected and extracted. Thus, they can also be seen as being

[9] `www.lixto.com`

subtree-selecting.

To facilitate learning, the queries are actually internally represented by tree automata rather than actual transducers. The automata define languages of *annotated trees*, i.e., trees in which each node is marked by a Boolean value, indicating whether it should be selected or not. Importantly, the languages defined must be functional in the sense that given a non-annotated tree, only one unique annotation is allowed, representing the choices of the query.

The authors present two algorithms, one for learning a node-selecting transducer from fully annotated data and one for learning interactively from partially annotated data. By fully annotated data, they mean a set of HTML documents in which all elements that should be selected by the extractor are marked (and no other elements are marked). Given such data, an RPNI-style[10] algorithm is employed to infer a deterministic tree automaton. The standard RPNI algorithm is modified slightly to make sure that the merging of stated is conditional not only on preservation of determinacy, but also on the preservation of functionality. Internally, the system works with *stepwise tree automata* [CNT04]. These are automata that work on the so-called curried binary encoding of unranked trees. The reason for this is that it is difficult to find a suitable notion of bottom-up determinism for unranked tree automata, particularly one that allows for unique minimization [MN07]. In a setting where ranked trees can be used, standard deterministic tree automata can be used instead.

As discussed previously, it is in most cases unrealistic to have access to sufficient amounts of fully annotated example data to facilitate learning of any reasonable quality. For this reason, the authors present an algorithm for learning the transducer from partially annotated data. The algorithm is in the spirit of Angluin's MAT-learning [Ang87]. It introduces so-called *Correct Labeling Queries*. The learner presents the teacher with a tree in which some nodes are selected. The teacher either answers that the annotation is correct or points to a node that is selected although it shouldn't be or a node that is not selected although it should be. There are also equivalence queries similar to those of Angluin.

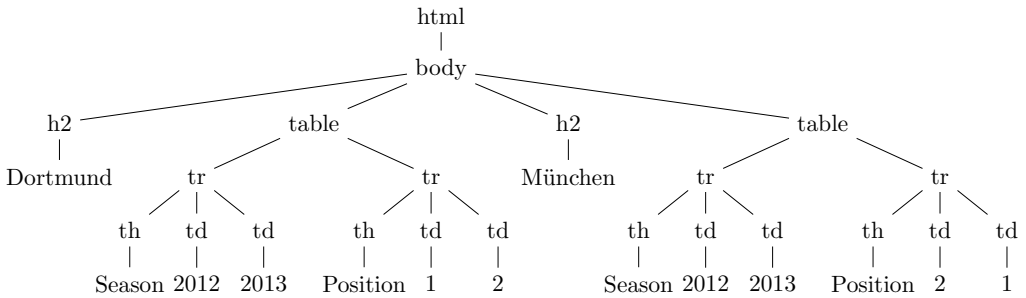
Rather than working with tables, as the original MAT-learning algorithm [Ang87] and previous variants for tree languages [Sak90, DH03], the algorithm gradually builds a selection of fully annotated trees and uses the previously discussed RPNI algorithm to construct hypothesis transducers.

The algorithm is implemented in the SQUIRREL information extraction system. Here, the end-user can design queries using a graphical user interface. She plays the role of the teacher. The system can ask Correct Labeling Queries by showing the user web pages where some elements have been selected. If the user is not satisfied with the result, she can mark one or more incorrectly labeled elements (i.e., missing or unwanted selections). The query hypothesis is then updated and the process is iterated.

Learning relations The relative simplicity of the SQUIRREL approach is attractive, even if the restriction to node-selecting queries is quite limiting. In an article from 2006 [GMTT06], Gilleron et al. expand on many of the ideas from [CGLN07] but focus on the extraction of *tuples* from semi-structured data. The authors note that many approaches to machine-learning of extractors for web data only handle the most straightforward cases of data organization, i.e., simple tables and lists. Other cases must be handled by post-processing. As an example of how data can be stored on a web page, see Figure 5. If

[10] Regular Positive and Negative Inference

Figure 5 An example of how relational data can be stored in a semi-structured setting. A (small part of) an imagined web page with results from the German Bundesliga.



the target relation is (TEAM, SEASON, POSITION), so that, e.g., the tuples (Dortmund, 2012, 1) and (München, 2013, 1) should be extracted, list- and table-based extractors are insufficient.

As in [GMTT06], the authors aim at an interactive wrapper inference system that is easy enough to allow non-technical users to design meaningful wrappers.

The main approach is to learn n-ary queries by starting with unary tuples and inductively adding one position at a time. The interactive setting for learning a query is similar to that of [CGLN07]. The user is first presented with an unannotated document and is asked to mark tuples that should be extracted. The system infers a query and shows the results of running the query on the document to the user, who is then prompted to supply false positives and false negatives. Once the user is satisfied with the result on one document, it is added to a collection of fully annotated example documents which is used in the ongoing interactive design, as the user is presented with new documents.

Questions of interest In the MICO setting, a number of interesting questions present themselves.

1. Can interactive learning of node selecting tree transducers be adapted for structure based NLP search?
2. How can interactive learning of tuple extractors be adapted to make use of cross-media data?
3. Can the existing learning settings be supplemented with statistical information in order to improve the effectiveness of interactive learning?
4. How well does the inherent structure in web pages and XML documents interact with the structure of extracted media metadata from the web page elements? Can tree-shaped meta-data be usefully integrated into the overall tree structure to allow interactive learning of cross-media extractors to be efficient?

2.4 Audio-Visual Analysis

2.4.1 Low-Level Visual Feature Extraction

Responsible partner/Author: FHG, Christian Weigel
Related Technology Enablers: TE-201, TE-210, TE-211

The extraction of visual low level features from images or videos is a key component for the selected MICO showcases. As pointed out in section 2.1 they establish the foundation for the retrieval of higher semantics from image and video data. They can be used for image and video similarity detection through feature matching as well as supervised or unsupervised visual classification tasks of low- and high level semantics. Specialized feature extractors and descriptors can be used for tasks such as species classification (see section 2.4.3) or face recognition (see section 2.4.4)

This section gives a summary of the latest findings in visual feature extraction. This field of research has be very active in the last decade leading to a vast number of methods and types. Therefore we will limit this section to features/descriptors that may be applicable to MICO show-cases. Finding a suitable taxonomy for this purpose is difficulty since a number of properties can be used to group the methods. Some of those parameters are their locality, the applied domain (spatial, temporal), the captured scene properties (e.g. texture, color, shape), the computational/memory effort for extraction and storage, their robustness against image distortion such as transformation (rotation, scale, affine transformation), noise or illumination changes. For the sake of simplicity we group them based on their locality property, that is, we will give an overview of *local* and *global* visual features extractors and descriptors.

First, we will need to clarify some terms. A *feature* is the abstract term for a (application case) specific property of an image or video. Usually this feature must represent that property as well as possible while being clearly separable from other features of the same domain. Images and videos are spatially and temporally discretized functions of real time scenes (or more precisely a light field) that are represented by *pixels*. This representation is usually extremely high dimensional when converted to feature vectors. The common aim of feature extraction is to compact that representation and to transform it into a domain space suitable for a specific task such as matching or classification. This transformation leads to a so called *descriptor* of the feature which can be either a local feature or a global one (representing the whole image or video).

Local Feature Detection and Description Local feature extraction is the analysis and description of features called *interest points* (also referred to as *key points*) or *areas* at specific positions of an image. The finding of these interest points or areas is always the first step of a local feature extractor and usually termed *detection*. The interest points / area must be salient in some way (well-defined position, unique texture around) and the localization of them must be repeatable. The reason for this is the fact that, ideally, interest points must be reliably matchable among images under different camera transformations and properties, lighting conditions etc.. Thereby, the images can be records of the same scene (as in stereo vision) or from a different scene as in object matching or face recognition. Depending on the content of the image and the algorithm used the number and position of such local features vary.

The underlying principle of all interest point detectors is the concept of corners. Based on image intensities or image gradient filter responses such corners are detected. In order to include robustness against scale or rotation of the image this search is done at different scales (in a so called scale-space). The position, scale and orientation property (again often estimated based on gradients around the interest point) become properties of the interest point. Applications, that do not need these robustness leave out scale and/or orientation estimation respectively which yield faster computation times.

Once the interest points has been estimated, a rectangular, circular or arbitrary region (*image*

patch) around is defined. This region may also be detect by other means than using interest points (see section 2.4.2), e.g. using blob detection. The region is used to build the descriptor which is again often based on gradient histograms, approximations using differently oriented box filters or binarization based on intensity thresholds. The difference of several proposed methods lies in the sampling pattern (structure, number) around the interest point, the method (and thus speed) of the calculation. Depending on the methods chosen, the estimated descriptor can be robust against noise, illumination changes, scale, rotation (i.e. similarity transformation) and is often robust against affine or even projective transformations of the image.

In 2004 David Lowe presented the local texture feature descriptor *SIFT* [Low04] which gained much attention in the research community. The presented detection and description method proved to be scale and rotation invariant and gained a high repeatability. Since then several new approaches have been presented. Some targeting speeding up the extraction process while keeping the quality high, some tried to reduce the memory requirements for the descriptors. Papers about new detection methods, new description methods and about a combination of both emerged. Yet, SIFT still serves as an anchor for best achievable quality.

Instead of giving a full text description of the most important feature detectors and descriptors, tables 1, 2, and 3 give a rough overview. Although describing the basic idea of each method, the reader is referred to the original publications for an in-depth explanation. We will limit ourselves in this section to interest point detectors. For the detection of areas (i.e. blobs, ridges etc.) such as in object, face or animal detection and their interdependency with interest point detectors please see section 2.4.2.

Name	Year & Reference	Detector method	Descriptor creation	Dimension & Comp. Effort	Robustness
SIFT (Scale Invariant Feature Transform)	2004, [Low04]	Creation of scale-space by Differences of Gaussians (DoG) to estimate keypoint location and scale by non-maximum suppression, sub-pixel positioning by Taylor expansion, edge response elimination, orientation assignment	Gaussian weighted gradient magnitude and orientation around key point, normalized orientation histogram in the selected region, vector element thresholding and normalization to unit length – can be interpreted as 3D-histogram	128 float (high)	Scale, rotation, noise, illumination
SURF / U-SURF (Speeded Up Robust Features / Upright SURF)	2006, [BTG06, BETVG08]	Approximated second order Gaussian derivatives as box filters on an integral image at different scales. Scale space created by increasing box filter size instead of image scaling and repeatedly Gaussian filter convolution, interest points by non-maximum suppression.	Orientation assignment (not U-SURF) using Gaussian weighted Haar-Wavelet responses applied to circular area around interest point (using integral image) which are summed in windows (pie slice) in two dimensional orientation space. Square is place at interest point and oriented according to main direction. Weighted and normalized Haar feature responses in 4x4 sub region of those squares form the descriptor vector (unit length).	32, **64**, 128 float (medium)	Scale, rotation(SURF only), noise, illumination
ORB (Oriented FAST and rotated BRIEF)	2011, [RRKB11]	Detection of FAST-9 interest points and removal (ranking) based on long edge response using Harris corner measure on an image scale pyramid. Orientation estimated by intensity centroid method.	Key point orientation is used to rotate the test locations and thus build a "steered BRIEF"-operator that takes the rotation (angle discretized to 12 degrees) into account. In order to increase the loss of variance caused by that, one time training is performed leading to rBRIEF.	256 binary (medium)	scale, rotation, noise
BRISK / SU-BRISK (binary robust invariant scalable keypoints, SU: single-scale, upright)	2011, [LCS11]	Uses FAST 9-16 mask. Score as saliency measure but applied on continuous scale space. Saliency maximum is sub-pixel refined and interpolated between 3 octaves (by fitting 2D quadric function to the three 3x3 score patches and then a 1D parabola along the scale-axis)	Creating a binary string by concatenating the results of simple brightness comparison tests. Circular sampling pattern with $N = 60$ points and Gaussian smoothing (with std. deviation proportional to distance) at each sample point used to estimate gradient at this position. Pattern is rotated according to gradients. Bit-vector created by pairwise comparison.	512 binary (Medium)	scale, rotation

Table 1: Local feature detector *and* descriptor methods

Name	**Year & Reference**	**Detector method**	**Comp. Effort**
Harris Corners	1988, [HS88]	Energy function based on intensity difference in shifted windows must be maximized. Using Taylor expansion, this equation can be transferred into a matrix function where gradient responses represent a so called structure tensor. The eigenvectors (or as simplification the determinant and trace) of the structure tensor decides whether a corner, an edge or a flat area is detected: Using a threshold and the local response maximum leads to the interest points.	high
CenSurE, SUSurE (Center / SpeededUp Surround Extremas)	2008, 2009, [AKB08] [EMC09]	Computation of simplified bilevel Laplacian of Gaussian using box filters and integral images. Weak responses are filtered. Local extrema are detected. To these extrema the Harris measure (see above) is used to detect corner responses. Can be applied on mulitple scales and combined with non-maximal suppression. SUSureE adds efficient calculations of box filter responses by introducing a thresholding in the response calculation to avoid a number of sums.	low
FAST (Features from Accelerated Segment Test)	2006, [RD06]	A circle (r=3.4 pixels) of sixteen pixels around pixel p of interest of which n contiguous pixels need either to be brighter than $I(p)+t$ or darker than $I(p)-t$ where t is a threshold. n is chosen to be 12 or 9 and high speed test applied before (only four compass directions). Exhaustive comparison is speeded up by machine learning using ID3 to generate a decision tree, non-maximal suppression on detected corners.	medium
AGAST (Adaptive and Generic Accelerated Segment Test)	2010, [MHB$^+$10]	Same corner detection approach as FAST but instead of training for specific scenes a dynamic adaptation process is employed. Using a binary decision tree and two more states that embodies value access cost (register, cache, memory) and pixel configuration probabilities. Adaptiveness achieved by switching between decision trees based on currently observed area (e.g. homogeneous or heterogeneous).	medium

Table 2: Local feature detectors

Name	Year & Reference	Detector method	Descriptor creation	Dimension & Comp. Effort	Robustness
PCA-SIFT	2004, [KS04]	SIFT	Pre-computation of an eigenspace to express the gradient images of local patches. Given a patch, compute its local image gradient. Project the gradient image vector using the eigenspace to derive a compact feature vector.	20, **36** float (High)	see SIFT (claims better robustness to noise)
GLOH (Gradient Location and Orientation Histogram)	2005, [MS05]	SIFT	Compute of the SIFT descriptor for a log-polar location grid with 17 location bins. The gradient orientations are quantized in 16 bins. This gives a 272 bin histogram. The size of this descriptor is reduced with PCA. The covariance matrix for PCA is estimated on 47 000 image patches collected from various images. The 128 largest eigenvectors are used for description.	128 float (High)	see SIFT (claims to be better in some areas)
FREAK (Fast Retina Keypoint)	2012 [AOV12]	arbitrary	Binary descriptor by thresholding DoG at receptive field pairs. Pairs to be used are determined by pre-processing training. Yields coarse to fine selection strategy (512 pairs) Orientation estimation using 45 pairs by summing estimated local gradients (BRISK like)	512 binary (medium)	rotation
BRIEF (Binary Robust Independent Elementary Features)	2010, [CLSF10]	e.g SURF, SIFT, FAST,...	Intensity thresholding of neighboring pixels in patch which has been smoothed previously (signs of derivatives) based on random test locations	128, 256, **512** binary (medium)	depends on detector
SKB (Semantic Kernels Binarized)	2011, [ZREK11]	SU-SURE	16 different 4×4 convolution kernels representing basic geometric structure (corners, edges, ridges, blobs and saddles) are applied to 16 positions (two schemes, support region can be either 12×12 or 16×16 pixels). Three kinds of binarization.	256, 512 binary (low)	scale (used for stereo matching)
LBP (Local Binary Pattern)	2006, [AHP06], see also sec. 2.4.4	Detected face / object region	Divide face image into rectangular (or other shape) regions, extract local histograms with LBP, concatenate histograms into descriptor, weighted matching possible. LBP: choose neighbor-hood (radius, sampling points), threshold gray scale value cmp. to center pixel (binary result). Interpret result CCW as decimal number, put into histogram.	var (medium)	Global illumination, some variants also to rotation

LTP (Local Ternary Pattern)	2011, [TT10], see also sec. 2.4.4	Detected face / object region	Compared to LBP, LTP knows 3-valued compare function quantizing to 1, 0, -1. Pattern is split into positive and negative patterns using only binary codes. Instead of using a regular (rectangular) grid to retrieve the histogram that is then χ^2 matched, LTP matches using the distance transform images of each LTP code value.	var (medium)	Noise, Complex illumination with pre-processing
HOG (Histogram of oriented Gradients)	2005,[DT05]	Detected face / object region	Gamma and color normalization. Gradient computations, Weighted vote into spatial & orientation cells with fine orientation coding(signed or unsigned gradient) and coarse spatial binning, contrast normalize overlapping spatial blocks, collect HOG's over detections window.	15120(high)	Illumination changes, noise
DAISY	2010, [TLF10]	region (or global application)	Build a SIFT-like 3D histogram but instead of calculating and using gradient vectors directly they are built by summing up by Gaussian convolution of the orientation map, sampling is done in a daisy like shape	var (medium)	Illumination change, noise, small rotation

Table 3: Overview of local feature detectors and descriptors

Global Feature Description A number of applications do not focus on specific regions or objects of an image or video but rather need to capture the *whole image or video content*. Typical cases are the matching of images or video sequences against each other or retrieving high level semantic concepts such as content (e.g. city, beach, woods), genre (movie, news, sports), or even mood (valence, arousal). Again, the task is to get the features that describe these properties as fast and as compact as possible. Since now the whole image or video content holds the information it is even more crucial to get compact description of spatially and temporally sampled pixel information, especially in the video case. In some application areas (matching) but also in context of the property, local and global features are not clearly distinguishable. For instance, a texture descriptor like HoG [DT05] might be applied to the whole image or to multiple patches of an image thus leading to a more compact presentation in the first case. In this section we will not give such an detailed review as for the local descriptors but rather explain the basic concepts and give some examples. We categorize the descriptors based on the feature they describe. These are color, texture and shape.

The earliest features mainly used for image retrieval used color information like color histograms [SWS$^+$00a, FBF$^+$94, PBRT99, SB91]. The color space is partitioned in bins and each pixel contribution to a specific bin is counted. Jeffrey divergence or Jensen-Shannon divergence (JSD) can be used to compare the histograms and to obtain a dissimilarity measure. The MPEG-7 standard also defines visual compact features based on colors. These are Dominant Color, Color Structure, and Color Layout. An overview is given in [Eid03, Ohm01, MOVY01]. In addition to simple color histograms they also capture the spatial distribution. Especially the Color Layout descriptor has a high discriminative power while being very compact (128 dimensional for an image). It represents the spatial color distribution using block-wise sampling and DCT transform. Color Corellograms [HKM$^+$97] also model spatial relationships by using pixel color distance corellograms but require much more computation than the Color Layout descriptor.

Tamura Features as described in [TMY78] combine features according to the human visual system: coarseness, contrast, directionality, line-likeness, regularity, and roughness. They can be summarized as texture features. Further texture descriptors (also defined by MPEG-7) are Homogeneous Texture, Edge Histogram, or Texture Browsing [Ohm01]. Also Gabor Features [PJW02](although also used locally for face recognition) belong to this group.

Mainly designed for image retrieval, some of the features mentioned above have been evaluated in [DKN08].

Basically, when used on every n-th frame of a video these feature can also be used for video retrieval and matching. Anyhow, they do not encode any temporal properties. Those can be captured e.g. by using the Optical flow [HS81] or a block based matching method and then combine (e.g. concatenate) the spatial and temporal features. Combined measures use a color layout like approach in the spatial domain but combine them temporally and use 3D-DCT transform followed by binarization to calculate segment wise descriptors [CSM06] just to give an example. There are several further methods we will not explain in detail here but rather give pointers to the interested reader [CZ03, KV05, CS08, LYK09, XHS$^+$10, TNC10, LLWH12, LLX13].

2.4.2 Object- and Animal Detection

Responsible partner/ Author: FHG/ Alexander Loos
Related Technology Enablers: TE-202

Autonomous detection of general objects and especially animals in images or short video sequences is a crucial component in selected MICO showcases. During the past 20 years a large and growing amount of literature describing techniques for different content-based image and video retrieval tasks has been published. Especially the retrieval of animal pictures has recently attracted the attention of computer scientists [Sch01a, BF06]. While Schmid *et al.* [Sch01a] solely exploit texture information in combination with unsupervised clustering techniques to retrieve images showing animals of the same kind, Berg and Forsyth [BF06] additionally incorporate other cues like color, shape, and metadata from Google's text search to identify images containing categories of animals.

A different field of research is the automatic detection of animals in images or videos. As opposed to image retrieval, animal detection aims at determining the locations and sizes of animals within an image or video. Automatic approaches for animal detection build the basis of subsequent tasks like tracking, behavior analysis, species recognition or individual identification. Many approaches that either detect segments of the animals' body (e.g. the face) or complete bodies in image or video sequences can be found in the literature. The following section gives an overview of state-of-the-art systems for automatic animal detection, tracking, and behavior analysis.

Automatic animal detectors can be classified into five main categories: *Template Matching*, *Rigid Object Detectors*, *Local Keypoint Detectors*, *Model-Based Detectors*, and *Motion-Based Detectors*.

Template Matching The simplest animal detectors use template matching to localize animals of a specific species in images or videos. One prototypical instance of the object's visual appearance is used to detected the animal of interest. A template image is typically generated by taking either only one or the mean of many example images that best represent the desired object. The template is then shifted across the source image. By calculating certain similarity metrics such as normalized cross-correlation or intensity difference measures [Cox95] the object location is defined as the area corresponding to the highest similarity score. Kastberger *et al.* [KMW+11] for instance apply a template matching technique to detect and track individual agents in densely packed clusters of giant honey bees. The authors use a 3D stereoscopic imaging method to study behavioral aspects of shimmering, a defense strategy of bee collectives. After detection, stereo matching, and tracking of individual bees, 3D motion patterns were analyzed using luminance changes in subsequent frames.

Although template matching algorithms might be fast and easy to implement, they only achieve adequate results in rather controlled settings and perform poorly in natural wildlife environments due to cluttered background and geometrical deformations of the object to be detected. To achieve invariance of the method against object deformations different scales and rotations must be applied which significantly increases processing time. Hence, more sophisticated object localization algorithms must be applied to automatically detect free-living animals in their natural habitats.

Rigid Object Detectors A more advanced group of detectors are called rigid object detectors. As the name suggests, these kind of detectors are limited to non-deformable objects with similar shape. Visual descriptors and the spatial relationship between them are utilized to localize rigid objects, where

features vary among instances but their spatial configuration remains similar. For human face and pedestrian detection, rigid object detectors have been used for over a decade. Although Rowley *et al.* [RBK98] already achieved promising results with a neural-network based human face detection system in 1998, the probably best known algorithm for real-time object detection was presented by Viola and Jones in 2001 [VJ01]. It uses a boosted cascade of simple Haar-like features which exploit local contrast configurations of an object in order to detect a face. The AdaBoost algorithm [FS99] is utilized for feature selection and learning. Numerous improvements have been proposed over the last decade to achieve wider pose invariance, most of them relying on the same principles as suggested by Viola and Jones. For an overview of face detectors the reader is referred to [HL01, YKA02, RRB12]. Later, Dalal and Triggs [DT05] proposed to use Histograms of Oriented Gradients (HOG) and a linear Support Vector Machine (SVM) for the detection of humans in low-resolution images. They showed that locally normalized HOG descriptors provide a powerful robust feature set to reliably detect pedestrians, i.e. humans in upright positions, even in cluttered backgrounds, difficult lighting conditions, and various poses.

After the successful application of these techniques to the field of human detection in natural scenes, computer scientists started to adapt and extend these ideas to detect animals in images and videos as well.

Burghardt *et al.* for instance proposed a system for lion detection, tracking, and basic locomotive behavior analysis in [BCT04, BC06]. Although the authors use an enlarged set of Haar-like features, the initial face detection stage is based upon the original approach introduced by Viola and Jones [VJ01]. Once a face has been localized in a frame, the Lucas-Kanade-Tomasi method [ST94] was applied to track the face region through the video sequence using a number of interest points on the lion's face. Furthermore, a rectangular interest model was created in locations a face was spotted to achieve accurate and temporal coherent tracking performance.

The approach of Viola and Jones [VJ01] has also actively been studied to detect heads of cat-like animals in images [ZST08, ZST11, KIKY09, SR12]. However, in [ZST08, ZST11] the authors argue that applying algorithms for human face detection directly to detect the head of cat-like animals would perform poorly. Since cat faces have a more complicated texture than humans faces and the shape of cat heads can vary significantly from individual to individual, the extraction of features would result in a high intra-class variance which is crucial for most detection algorithms. Zhang *et al.* overcome this burden in [ZST08] by jointly utilizing texture and shape by applying different alignment strategies to separately train a shape and a texture detector. The shape information is kept by aligning the cat face such that the distance between the two ears is the same through the entire training set. On the other hand by aligning the faces according to their eyes the texture information is preserved while the shape of the cat's head is blurred. Furthermore, they use a set of Haar-like features on oriented gradients as features which was shown to outperform other descriptors commonly used for face detection.

In a second step they jointly train a final classifier to fuse the outputs of the initial shape and texture detectors. In the detection phase first both detectors are applied separately by using a sliding window to get initial face locations. A final decision is made by applying the joint shape and texture fusion classifier. The same authors apply their algorithm to other cat-like species like tigers, cheetahs, and pandas in [ZST11]. To further handle the misalignment cost between both detectors they present a novel deformable detection approach which considers both misalignment cost and the outputs of the shape detector and texture detector. Also Kozakaya *et al.* use a two step approach to detect cat faces in images [KIKY09]. Opposed to the work done by Zhang *et al.* they do not use two different alignment

strategies but rather extract complementary feature sets to gather shape and texture information. In a first step a candidate search is performed which uses simple Haar-like features and AdaBoost as done by Viola and Jones in [VJ01]. Fast to compute and easy to implement but not discriminative enough to deal with complicated shape and texture, the authors suggest to use more sophisticated features in a second phase to verify face candidates. Therefor, Kozakaya *et al.* use co-occurrence histograms of oriented gradients (CoHOG) [WIY09] since they have strong classification capability to represent various cat face patterns. Due to the high dimensionality of CoHOG a simple linear classifier obtained by a linear SVM is used for candidate verification. They evaluate their approach on the dataset provided by Zhang *et al.* in [ZST08]. However, the experimental conditions are not strictly the same since the evaluation paradigm differs from the one used in [ZST08]. Nevertheless, the results suggest that the proposed approach outperforms the method by Zhang *et al.* significantly. Another advantage of this method is that the approach is much more generic since it is not restricted to detect faces of cat-like animals only but other rigid objects as well.

Starting from the assumption that humans and our closest relatives share similar properties of the face, rigid object detectors have recently also been used to detect African great apes [EK11]. The method by Ernst and Küblbeck is based on ideas of the approach by Viola and Jones [VJ01] but was improved significantly by using multiple consecutive classification stages with increasing complexity and different illumination invariant feature sets to detect faces of chimpanzees and gorillas in images and video sequences. Although the proposed system has some robustness to difficult lighting situations, it lacks in robustness to severe occlusion and far-off frontal poses. However, for subsequent analysis such as individual identification for instance, faces often are expected to be in a full-frontal pose. In addition to real-time capable face detection in images and videos, the authors also propose methods to automatically distinguish between chimpanzees and gorillas. The first approach uses the detection scores of the applied face detection models for chimpanzees and gorillas. For the second method a separate classification model is trained based on structure features only and applied to the detected face. Both techniques for species classification perform remarkably well with over 90% accuracy.

Although researchers use rigid object detectors mainly to localize the faces of animals, this class of object detectors has also been applied to detect whole animal bodies. Miranda *et al.* for instance use the face detection paradigm by Viola and Jones [VJ01] for the visual detection of bumblebees [MSV12]. Furthermore, the authors apply a discriminative tracking algorithm proposed by Gu and Tomasi [GZT10] to improve the detection in video sequences and fill the gaps where detection has failed. Only one test sequence with restricted variety of different backgrounds and just one single individual present has been used for experimentation which is not enough for a thorough evaluation. Nevertheless, the achieved results are promising for a preliminary study. Although rigid object detectors can be used to detect insects such as bumblebees due to the limited number of their body appearances, the localization of deformable objects in natural scenes requires more sophisticated techniques because different body postures would result in different appearances of the same object.

Model-Based Detectors To cope with the above mentioned challenges, researchers recently proposed model-based methods to reliably detect animals in visual footage. Different feature-sets can be used to create models of an animal body such as appearance, texture, shape, color or the combination of those.

Stahl *et al.* reported a preliminary study on the combination of multiple 2D and 3D sensors to capture biometric data of farm animals under real-life conditions in [SSH12] . More specifically the authors suggest to use a sophisticated hardware setup of two high-resolution monochrome cameras

and one integrated color camera to reliably detect the heads of horses in a livestock farm. Due to the constrained environment at the feeding area and the fact that the horse's head points towards the camera, a simple foreground-background separation algorithm based on the depth information provided by the stereo cameras can be used to obtain a coarse location of the horse. To distinguish the animal's head from the rest of the body an ellipse-like head model is subsequently created and fitted to the borders of the original head mask. This procedure builds the basis to locate, measure, and identify the animal by fusing information of multiple cameras. By crucially evaluating the proposed framework, the system has proven to perform well. However, the application scenario as well as the controlled conditions in a livestock farm allows a multiple camera setup and a relatively simple geometrical approach to achieve an excellent detection performance. Reliable detection of animals in more challenging natural environments however requires more sophisticated solutions.

Remarkable ideas for a generalized unsupervised object tracker and detector were presented by Ramanan and Forsyth in [RF03, RFB05, RFB06]. In [RF03] the authors propose a technique to automatically build models of animals from a video sequence where no explicit supervisory information is necessary and no specific requirements regarding background or species are stipulated. Animal bodies are modeled as 2D kinematic chains of rectangular segments where the topology as well as the number of segments are not known a priori. In a first step candidate segments are detected using a simple low-level local detector by convolving the image with a template which corresponds to parallel lines of arbitrary orientations and scales. This step alone results in many false positive detections. Therefore, resulting segments are clustered in a second step to identify body limbs that are coherent over time. After pruning away remaining segments that do not fit the motion model because the tracks are to short or move to fast, a coarse spatial model of the remaining segments can be assembled. Because appearance models of animals are generated on-the-fly there are two ways to think about the proposed system. It can either be seen as a generalized tracking system that is capable of modeling objects while tracking them or as a source of an appearance model which can later be used to detect animals of the same species.

One drawback of this approach is that when building the rough shape model of an animal's body one has to be certain that within a given sequence only one animal is present and it is the only animal in the scene. Furthermore, the resulting appearance model is very much tuned to the specific species in the video sequence. Therefore, the same authors try to overcome these drawbacks by extending their work towards an object recognition framework to detect, localize, and recover the kinematic configuration of textured animals in real-world images [RFB05, RFB06]. Ramanan *et al.* fuse the deformable shape models learned from videos and appearance models of texture from labeled sets of images in an unsupervised manner for that purpose. Although the detection results improve significantly compared to their previous work the detector is designed to only detect highly textured animals like tigers, zebras, and giraffes. The approach would fail for a majority of camouflaged or non-textured animals like elephants or great apes because detecting only vertical and horizontal lines to build the deformable model would result not only in false positive but more crucial false negative detections.

More recently, deformable part-based models (DPM) were introduced by Felzenszwalb *et al.* [FGMR10] as generic object detectors achieving state-of-the art results on a variety of object categories in international benchmarks such as the PASCAL VOC 2010 [EVGW$^+$10]. Whilst the majority of methods for object detection depict the object of interest as a whole, the main idea of DPMs is to represent objects as flexible configurations of feature-based part appearances. Support Vector Machines

(SVM) are used to learn the set of appearance detectors and part alignments. While performing very well on some object categories, DPMs are known to be sensitive to occlusion and highly deformable object categories such as animals. Parkhi *et al.* [PVJZ11] extend the ideas of DPM to distinctive part models (DisPM). They propose to initially utilize DPMs to detect distinctive regions such as the head of animals in the first phase. Secondly, the whole body of the animal is subsequently found by learning object specific features such as color or texture from the initially detected image region. After coarse foreground-background separation based on the learned low-level image features, graph cuts [BV01] are used for the final segmentation of the animal's body. Parkhi *et al.* apply their algorithm to detect cats and dogs in still images achieving results comparable to the state-of-the-art for other object classes. Furthermore the authors also claim that this technique can be used for a variety of other animal species as well.

Another study by Sandwell and Burghardt [SB13] uses three different models of DPMs to detect chimpanzee faces under difficult conditions such as far-off frontal poses or partial occlusion. Whilst the first and the second model are trained on the face region and an expanded version of the face region respectively, the third model integrates multiple spatially distributed DPMs. Different from the approach presented by Ernst *et al.* in [EK11] the proposed algorithm is not real-time capable. However, the authors conclude that the reduced reliance on facial features alone and the combination of the three proposed models has led to a detector which is far less sensitive to non-frontal poses and more robust to less well resolved faces as well as partial occlusions.

Local Keypoint Detectors A large and growing body of literature investigated template matching, rigid object detectors as well as model-based approaches to detect animals in images or videos. However, they often perform poorly when detecting animals in their natural habitat due to a wide variety of postures, lighting conditions and partial occlusion. Tiny object regions on the surface of an animal's body often exhibit less deformation than the entire organism [KB13]. Therefore, new approaches for fast, robust, and reliable detection and description of regions or local interest points such as the Harris Corner Detector [HS88], Scale Invariant Feature Transform (SIFT) [Low04] or Speeded Up Robust Features (SURF) [BETVG08] have been developed in the recent past. Inspired by the great success of local keypoint detectors and descriptors for object localization, matching, and categorization, these approaches have also been used for visual animal biometrics. The applications include a variety of species, ranging from insect categorization [LDZ$^+$07] to turtle identification [dZPR$^+$10] and other coat patterned animals like penguins and zebras [Bur08, BC10].

In 2010, de Zeeuw *et al.* proposed an approach based on SIFT matching for turtle detection and identification [dZPR$^+$10]. According to Zeeuw *et al.* leatherback sea turtles carry a so called *"pink spot"* on the dorsal surface of their head which is unique between individuals and can therefore be used for identification. Since the proposed algorithm is a semi-automatic approach, the first step is to manually crop the desired head region out of the original image. The extracted image patch is then compared with reference images using the basic SIFT matching approach originally proposed by [Low04]. The evaluation results on two challenging real-world datasets confirm the effectiveness and reliability of the algorithm proposed by Zeeuw *et al.* However, manual interaction is still necessary to annotate the region of interest. Thus, designing a detection algorithm that automatically locates the pink spot on the turtle's forehead is highly desirable.

For the proposed approaches by Larios *et al.* [LDZ$^+$07] and de Zeeuw *et al.* [dZPR$^+$10] local keypoint detectors serve as a pre-processing step to locate stable points of interest for the subsequent

extraction of discriminative information around each point. These descriptors are then used for individual identification, species classification or comparable tasks. Therefore, these approaches internally presume that the animal of interest is actually present in the processed image.

In [Bur08] Burghardt utilizes keypoint detectors as initial detection stage to robustly detect coat patterned animals. Keypoint locations are initially stipulated by traditional corner detection algorithms based on the auto-correlation matrix A of the input image over a small neighborhood. Since corner locations are defined by significant signal change in all dimensions, the two eigenvalues of A are analyzed to accurately detect corners in an image. More specifically, if both eigenvalues λ_1 and λ_2 have large positive values, then a corner is supposed to be found. This procedure is utilized in many corner detection algorithms such as the *Harris corner detector* [HS88] or the *Shi-Tomasi corner detector* [ST94]. In a subsequent step the area around a detected point of interest is described by placing a neighborhood window around the keypoint. A class-specific point-surround classifier is learned by extending the Viola-Jones framework [VJ01]. Instead of heuristically choosing the resolution of the neighborhood window, the dominant spatial frequency in coat patterns is utilized to estimate a suitable window scale. Furthermore, a form of supervised *bootstrapping* is used to increase the robustness of the detector against false positive detections. Other modifications of the original implementation of the Viola-Jones framework refer to perspective constraints and dense belief maps, e.g. real-valued classification outputs instead of binary decisions. For details the interested reader is referred to [Bur08]. Burghardt applies the proposed algorithm to detect frontal chests of Penguins, faces of lions, and hindquarters of zebras achieving results comparable to state-of-the-art face detection results on humans. At a false positive rate of $4 \cdot 10^{-3}\%$ the detector achieves a detection rate of over 96%. However, false positive detections predominantly occur for highly cluttered background in natural environments, challenging lighting conditions and hard shadows, as well as cryptic resemblance due to groups of patterned animals imitating regional patterns of a single individual.

Although local keypoint detectors are capable of robustly detecting and describing points of interest in certain scenarios, they disregard important global information such as body structure, spatial relationship between keypoints, and temporal information available in video sequences. Moreover, local keypoints can only be used for coat pattern animals or species with distinctive natural markings on fur or skin. However, many species, such as great apes like chimpanzees and gorillas for instance, do not carry obvious natural markings which can be used for detection.

Motion-Based Detectors Although the above mentioned approaches achieve sufficient results under certain conditions, they lack in taking full advantage of the spatio-temporal information in video sequences to detect moving objects. Based on the assumption that even camouflaged animals can be detected by taking their movement in front of relatively stationary backgrounds into account, a considerable amount of literature has been published on motion-based detectors.

Especially the problem of detecting and tracking marine animals in underwater video has been tackled by many researchers [WEK04, ECD+06, SPB+12, KP12]. Either remotely operated underwater vehicles (*ROVs*) or live video feeds from stationary installed underwater cameras are used by biologists in order to perform marine ecological research within the *Fish4Knowledge*[11] project for instance. However, according to [KP12] footage gathered in unconstrained underwater environments often bring state-of-the-art object detection algorithms to their limits due to cluttered and periodically

[11] `http://fish4knowledge.eu` Last visit: August 8th, 2013

moving background, permanent lighting changes as well as the degrees of freedom marine animals can move.

To overcome this issue Walther *et al.* [WEK04] proposed a system capable of automatically detecting and tracking objects of interest in underwater video. Due to the fact that simple contrast-based detection algorithms are prone to a number of effects present in underwater video footage, such as non-uniform lighting conditions for instance, a background subtraction algorithm builds the first step of the framework. By subtracting the current frame from the mean image of at least 10 frames for every color channel separately, even translucent foreground objects can be separated from the background sufficiently. The actual object detection is subsequently applied using a saliency-based detection approach originally published by [IK99]. Furthermore, the authors found that oriented edges are a extremely useful features to detect marine animals and distinguish low-contrast translucent animals from organic debris. Once an object has been detected, Kalman filters [Kal60, WB06] are initiated to track the centroid of the detected objects. In a post-processing step object tracks that are shorter than at least five consecutive frames are discarded as noise. A performance evaluation of the proposed system on a single frame basis was conducted for two different data sets as well as a 10 minute video and achieves promising results with detection rates up to 80%. However, the question arises how many false positive detections were made by the system. Moreover, especially the evaluation on the detection and tracking performance is limited because the test set contains only a single 10 minute video. Two years later the same authors extended their approach in [ECD$^+$06] where a more thorough evaluation of object detection and tracking modules was conducted and the results of their previous paper were validated. Furthermore, the authors present a technique to subsequently classify the detected objects into biological taxonomies. For each tracked object a feature vector based on Schmid invariants [SM97] was extracted and a Gaussian Mixture Model (GMM) was used for classifying the three most common classes. Although species classification was at a very early development stage at that time, promising results were achieved for the three examined animal categories.

A quantitative performance evaluation of object detection algorithms in underwater video footage can be found in [KP12]. The authors compare and thoroughly evaluate several state-of-the-art object detection algorithms for their application to detect moving objects in underwater settings. The used dataset for evaluation was taken from the publicly available *Fish4Knowledge* projects database. The performed experiments suggested that an algorithm called *Video Background Extraction* [BvD11] outperformed the other approaches and therefore is most robust against the above mentioned challenges. However, the performance of all algorithms significantly decrease if typhoon events or storm is present in the video sequences.

Most recently, Spampinato *et al.* described a framework for automatically analyzing fish behavior during typhoon events in real-life underwater environments [SPB$^+$12]. Different texture and blurring features in combination with machine learning algorithms are used to detect typhoon or storm events in videos in a first step. In the second component of the proposed system fish detection and tracking is performed. Similar to the work of [KP12] four algorithms have been implemented and compared against each other to detect fishes under extreme conditions. According to the authors each approach perform fairly well were *Intrinsic Models* [Por05] performed best. However, often false positives are detected that have to be filtered out during post-processing. To deal with that problem additional features, such as color difference and difference of motion vectors at the object boundary as well as internal motion and color homogeneity, were extracted in a post-processing step and merged into a quality score. Only objects whose quality scores exceed a pre-defined threshold are

considered for further processing. Once the desired objects were detected, trajectories are extracted using a covariance-based tracking algorithm [POM06] which is known to handle the typical challenges of tracking objects in underwater environments well [Por06]. Within a last step, the extracted 2D trajectories as well as the object size, which indicates movement in the third dimension, are analyzed to evaluate movement patterns and behaviors of fish during typhoon events. Each module of the proposed framework was evaluated on a sufficiently large data set of 257 video sequences of 10 different cameras.

Beside approaches to detect marine animals in underwater environments, a growing body of literature has investigated motion-based detectors to localize mammals and birds in videos.

In 2009, Wawerla *et al.* [WMM+09] reported an automated wildlife monitoring system called *BearCam* which was deployed near the arctic circle to detect grizzly bears in videos. The system is located at a river site and monitors the animal's feeding behavior for four hours per day. To assist biologists with tedious annotation work, Wawerla *et al.* developed an algorithm for automatic detection of bear appearances in recorded video. The authors extend the shapelet features by Sabzmeydani and Mori [SM07] by additionally incorporating motion information from gathered video material. Shapelet features are a set of sophisticated mid-level features, originally developed to detect pedestrians in still images. They are constructed out of low-level gradient information using the AdaBoost learning algorithm [FS99]. In addition to simple gradient information as low-level descriptors Wawerla *et al.* exploit background differences, computed by taking the median over a sampled set of frames. Motion shapelet features are then constructed as a weighted combination of the previously extracted low-level gradient and background information within a specified sub-window using AdaBoost. In a third and final step again AdaBoost is used to combine the information of different regions across the image and thus build the final classifier. A commonly used sliding window approach is used in the detection stage to localize the appearance of a bear in every frame. Extensive experiments proved the usefulness of the proposed algorithm which achieved suitable detection results. However, many false positive detections were found in highly textured regions and in areas with large amount of motion, e.g. at the banks of the river or regions of water. Moreover, because the detection is performed on every single frame the system is not real-time capable. Object tracking algorithms could speed up the performance while at the same time boost the detection accuracy of the system because non-moving objects may be removed in a post-processing step.

Song *et al.* described a robust autonomous system that assists ornithologists in observing and cataloging flying birds in [SQX+06]. Autonomous high-resolution video cameras were installed in the field which continuously scan the sky and automatically detect birds flying in the field of view. Video frames in which birds were detected are automatically send to the ornithologists for further processing. However, to save computational complexity and processing time only every fourth frame is scanned for birds using a non-parametric motion filtering technique proposed by Elgammal *et al.* in [EHD00]. Similar to the background subtraction algorithm used by [KHWB05], the method uses a Gaussian model to distinguish moving objects from constant background. Because the Gaussian distribution updates itself when a new sample comes in, periodic movements by branches or trees can be characterized by the model. Because this technique alone would result in too many false positive detections due to cloud movements and other non-periodic motions, temporal difference filtering is used to estimate the velocity of detected objects in adjacent frames. Due to the fact that birds usually move a lot faster than clouds, false positive detections can be ruled out to a certain degree. Although the approach is rather simple, the authors found that during a long-term study of 310 days where videos were captured continuously, 99.9953% of the data to be sent to the ornithologists could be removed

by the proposed algorithm. However, because the system was designed to have a low false negative rate still 96% of that data was due to false positive detections. Moreover, the authors were unable to present a performance measure for the missed detections. Admittedly, Song *et al.* tried to measure the false negative rate using a two hour video. However, no bird was missed by the system in this single video file. Yet, because only every fourth frame is scanned thoroughly, it is very likely that birds may be missed in the long run.

In 2012, Khorrami *et al.* published a paper in which they describe a system for the detection of multiple animal species in low frame-rate videos typically used by biologist to autonomously gather camera trap videos in wildlife environments [KWH12]. The authors use a recently developed technique for foreground-background separation called Robust Principal Component Analysis (RPCA) [CLMW11]. RPCA splits each frame of a video sequence in a low-rank matrix L which contains pixels of the background and a sparse matrix S representing the foreground activity of moving objects. An occurring animal is then isolated from the remaining foreground by calculating the local entropy for a small neighborhood around every pixel. While areas with similar intensity will have low entropy values, abrupt changes due to edges caused by the boundary of the animal's body correspond to high entropy. Since this procedure still results in a relatively high number of false positive detections, the Large Displacement Optical Flow algorithm by [BM11] detects large changes of velocity by incorporating motion information. The region with the highest amount of motion is considered to be the animal to be detected. Although the proposed method achieves promising results on a realistic dataset of ten different animal species, a high number of false detections occur in sequences with a high degree of background motion caused by rain and snow for instance. Another major drawback of the approach is that only one single individual at a time can be detected within a frame since only the candidate segment with the highest motion is considered to be the animal.

Most recently a system for automatic detection and tracking of elephants in unconstrained wildlife videos was proposed in [Zep13]. Zeppelzauer *et al.* argue that current state of the art systems for animal detection and tracking often explicitly focus on highly textured animals. However, for animals with poorly textured skin like elephants for instance other visual cues must be investigated. Also shape features would be impractical for the detection of animals due to pose variation and partial occlusion in natural habitats. Therefore, the authors propose a method that learns a color model of elephants from few labeled training data. In a first step a mean-shift clustering algorithm [CGM02] is used to extract spatial segments of the same color. Based on labeled training data a Support Vector Machine (SVM) is trained to distinguish background color from foreground color in the LUV color space. However, as claimed by the authors, color alone is a weak and unreliable feature for elephant detection since many objects in a natural environment have similar colors as elephants which leads to a unreasonably high number of false positive detections. To overcome this issue, the authors efficiently exploit temporal information to significantly reduce the number of false positives. Each initially detected segment is subject to a tracking algorithm based on the optical flow of the segment's pixels. Tracked segments are then joined into sets of coherent spatio-temporal candidate segments, i.e. segments belonging to the same objects are connected. Based on a number of extracted spatio-temporal features like the tracking duration and changes of the segment's shape, the final decision of the appearance of an elephant is made. Since tracking of segments establishes temporal relationships over several frames, missing detections can be interpolated and tracking gaps are closed in a post-processing step.

The proposed system was evaluated on a realistic dataset of elephant videos gathered under real-life conditions by biologists and achieved high detection accuracy of 90% at a low false detection rate

below 5%. Since the approach is claimed to be insensitive to pose variation, lighting conditions, and the number of individuals present in the video sequence, the authors did not have any further requirements for the tested video material. However, researchers and gamekeepers often use infrared cameras in order to monitor animals during night. Therefore, a different approach as initial object localization must be investigated in order to process gray-scale and infrared footage as well.

As could have been seen a variety of different approaches exist to reliably detect animals in images and videos. Table 4 summarizes this section and compares the reviewed algorithms.

Category	Reference	Species	Description	Notes
Template Matching	Kastberger *et al.* [KMW+11]	Honey Bees (body)	(1) 3D stereoscopic imaging (2) Analyzing basic behavioral patterns (3) Detection and tracking of individuals	(1) Sophisticated hardware setup necessary (2) Template matching only achieves adequate results for species with non-deformable bodies
Rigid Object Detectors	Burghardt *et al.* [BCT04]	Lions (faces)	(1) Viola-Jones AdaBoost cascade (2) Enlarged set of Haar-like features (3) Lukas-Kanade tracker of interest points (4) Basic locomotive behavior based on vertical head movement	Behavioral analysis could be improved by taking spatio-temporal interest points of the whole body into account
	Zhang *et al.* [ZST08, ZST11]	Cat-like animals (faces)	(1) Jointly utilizing shape and texture using different alignment strategies (2) Haar-like features and gradients used as features (3) Fusing the results of separately applied detection models for final decision	(1) Extension of the algorithm to detect not only cats but also faces of cat-like animals (tigers, cheetahs, pandas) (2) Application restricted to cat-like animals only
	Kozakaya *et al.* [KIKY09]	Cats (faces)	(1) Opposed to [ZST08, ZST11] different feature sets are used to represent shape and texture simultaneously (2) Haar-like features and AdaBoost for candidate search (3) CoHOG and linear SVM for validation	(1) Outperforms approach by [ZST08, ZST11] (2) Claimed to be more generic than [ZST08, ZST11]

Category	Reference	Species	Description	Notes
	Ernst *et al.* [EK11]	Great Apes (faces)	(1) Real-AdaBoost with multiple classification stages and increasing feature complexity (2) Gradient, structure, and census features used in different stages (3) Multi-resolution approach using image pyramids (4) Species classification based on detection confidences (5) Face tracking in video using Kalman filters	(1) Suited for detection in still images as well as videos (2) Tracking of detected objects through video sequences (3) Lacks robustness to far-off frontal poses and severe occlusion (4) Used in this thesis as initial detection and tracking algorithm for subsequent identification
	Miranda *et al.* [MSV12]	Bumblebees (body)	(1) Viola-Jones AdaBoost implementation in combination with Haar-like features (2) Body tracking using Gu-Tomasi tracker [GZT10]	(1) Limited experimental verification (2) Treats insect bodies as rigid objects due to small pose variations
Model-Based Detectors	Stahl *et al.* [SSH12]	Horses (head)	(1) Multiple 2D and 3D sensors (RGB and infrared cameras) (2) Coarse head localization using depth information (3) Fitting an ellipse-like head model for refinement	(1) Used for detection in livestock farms (2) Constrained application environment (head has to point towards the cameras) (3) Sophisticated hardware setup necessary
	Ramanan *et al.* [RF03, RFB05]	Various textured animals (body)	(1) Generalized unsupervised object detector and tracker (2) Candidate segments detected using low-level texture detector (parallel lines) (3) Animal bodies are modeled by a 2D kinematic chain (4) clustering of segments to identify body limbs that are coherent over time	(1) No species-specific descriptors necessary for detection (2) Only one individual has to be present in the video (3) System is designed for highly textured animals only (4) Algorithm can be seen as a generalized object tracking system or as source for appearance model for detection

Category	Reference	Species	Description	Notes
	Parkhi *et al.* [PVJZ11]	Cats and dogs (body)	(1) Based on DPMs as initial distinctive region detectors (2) Detected region is used to learn species specific low-level features (3) Graph cuts [BV01] utilized for final body segmentation	(1) Approach outperforms DPMs for detection of animals (2) Performance could be further improved by incorporating further cues for learned appearance model
	Sandwell *et al.* [SB13]	Chimpanzees (faces)	(1) Three different detection models (2) Integration of multiple spatially distributed DPMs	(1) Detection of non-frontal and occluded faces possible (2) Not real-time capable at this time
Local Keypoint Detectors	de Zeeuw *et al.* [dZPR$^+$10]	Leatherback Sea Turtle (head region)	(1) Detection of local interest points (SIFT [Low04]) at the "pink spot" located on turtle's head (2) Basic SIFT-matching [Low04] for identification (3) Verification based on affine transformation and gray-level pixel intensities	(1) Initial manual segmentation of area of interest necessary (2) SIFT features mainly used for identification
	Burghardt *et al.* [Bur08, BC10]	Penguins (body) Lions (face) Zebras (body)	(1) Initial detection of interest points based on eigenvalues of image autocorrelation matrix (see e.g. Harris corner detection [HS88]) (2) Learning of species-specific point-surround classifier by extension of Viola-Jones framework [VJ01]	(1) High detection rates up to 96% (2) False positive detections mainly caused by highly cluttered background, hard shadows, and cryptic resemblance
Motion-Based Detectors	Walther *et al.* [WEK04] Edington *et al.* [ECD$^+$06]	Marine animals (body)	(1) Initial object localization using background subtraction (2) Validation by detection of salient regions and oriented edges (3) Kalman filters for object tracking (4) Post-processing to further erase false positives	(1) Promising detection results up to 80% (2) As shown in [KP12] more sophisticated motion-based object detection algorithms may further increase the performance

Category	Reference	Species	Description	Notes
	Spampinato *et al.* [SPB+12]	Marine animals (body)	(1) Comparison of several motion-based object detection algorithms (2) Difference of motion vectors and color features to further eliminate false positives in a post-processing step (3) Covariance-based tracking (4) Analysis of movement patterns during typhoon events	(1) Detection performance of all algorithms decrease significantly during storm or typhoon events (2) Behavioral analysis only for groups of fish not for individuals
	Wawerla *et al.* [WMM+09]	Grizzly bears (body)	(1) Background subtraction based on frame differences (2) Extension of shapelet features [SM07] to motion shapelets to improve detection performance (3) Sliding window approach used for final detection	(1) High number of false positive detections in cluttered regions and segments with high motion (2) Detection is performed in every frame (3) Tracking algorithms could increase the performance in terms of speed and accuracy
	Song *et al.* [SQX+06]	Birds (body)	(1) Non-parametric motion filtering based on GMM (2) Elimination of remaining false positives using temporal difference filtering	(1) Amount of data to be manually analyzed by ornithologists could be decreased significantly (2) Many false positive detections caused by background motion (3) Number of missed detections remains unclear
	Khorrami *et al.* [KWH12]	Multiple animal species (body)	(1) RPCA for foreground-background separation (2) Local entropy over small regions in combination with large displacement optical flow for refinement	(1) False positive detections caused by high degree of motion due to rain or snow (2) No restriction to a certain species (3) Only one animal at a time can be detected

Category	Reference	Species	Description	Notes
	Zeppelzauer *et al.* [Zep13]	Elephants (body)	(1) Learned color model and SVM for foreground-background color classification (2) Optical-flow based tracking (3) Post-processing to decrease number of false positives	(1) Algorithm is claimed to by insensitive to pose variation, lighting, and number of individuals (2) Algorithm would fail for gray-scale and infrared video footage often used for animal monitoring

Table 4: Overview of state-of-the-art algorithms for animal detection in images and video footage.

2.4.3 Species Classification

Responsible partner/ Author: FHG / Alexander Loos
Related Technology Enablers: TE-203

After the automatic detection of animals in images and videos automatic approaches for species classification can be applied. It is therefore crucial to review state-of-the-art techniques for automatic species recognition since they are important for selected MICO showcases.

Besides individual identification of animals in audio-visual footage, the development of automatic routine procedures to reliably classify the species of detected animals arose interest of researchers and computer scientists within recent years. Taxonomic classification of animals is a prerequisite for many biological questions such as biodiversity conservation and natural resource management [SWJ+09]. However, to date only few automatic routine procedures have been proposed in the literature to classify the species based on their morphological traits. The development of computer-vision algorithms to automatically identify the species of an animal is not a trivial task because

(1) individuals of a given species may differ drastically in their morphology due to phenotypic variations caused by age or environmental conditions and

(2) different species may have similar morphological traits because many taxonomic groups often comprise thousands of species [GO04].

An enormous amount of progress has already been made in the field of automatic insect classification. Three of the most promising systems which are frequently used by experts to reduce the burden of manual classification of specimens are *ABIS* [Ste00], *SPIDA-web* [RDHP07], and *DAISY* [O'N10]. Within this report a coarse outline of these systems as well as more recent approaches is given. For a state-of-the-art overview of the above mentioned animal classification frameworks the interested reader is referred to [GO04, SWJ+09, LDZ+07, Mac07].

The *Automated Insect Identification System (ABIS)* was one of the first sophisticated approaches for automatic taxonomic categorization of bees based on the venation of their wings [Ste00, ASSW01, Sch01b]. Each bee is manually positioned under a microscope with background illumination in standard pose. For classification, the system follows a hierarchical approach by first determining a set of key wing cells, the area between veins, based on line and intersection detections. A set of low-level descriptors is subsequently extracted for initial classification in order to select a certain pre-defined deformable venation template saved in an external database. Once the template is fitted to the wing image, remaining cells can be reliably detected and the previously extracted feature vector is extended with a number of features obtained from the intensity values within a sampling window [ASSW01]. A Support Vector Machine (SVM) and Kernel Discriminant Analysis (KDA) are finally used for classification. The system is known to perform well even for bee species that are known to be hard to classify even for human experts. Although the features used to classify bee species are known to perform well, they make the system very specialized to a certain kind of insect. Another system commonly used by experts to distinguish between different spider species is called *Species Identification, Automated and web-accessible (SPIDA-web)*[12], introduced by Russel and Do in [RDHP07, DHN99]. The proposed algorithm utilize artificial neural networks (ANN) to

[12] `http://research.amnh.org/iz/spida/common/index.htm` Last visit: January 17^th, 2014

classify spiders based on their external genitalia. Direct user interaction is required to annotate the region of interest within an image gathered under a microscope with constant background lighting. The wavelet coefficients of Gabor filters are used as input for multiple back-propagation neural networks trained for each species separately. Preliminary results for female spiders presented in [RDHP07] indicate that *SPIDA-web* is capable to achieve accuracies up to 95%. Although it was claimed by the authors that the system was created as a generalized classification system it was to date only tested on spiders. To overcome these limitations O'Neill *et al.* proposed a generic identification system based on pattern recognition called *Digital Automated Identification System (DAISY)*[13] in [O'N10, O'N07]. While the first version of *DAISY* [OGGW00] exploited the Eigenfaces approach originally developed for human face recognition [TP91a], the core classification algorithm of the recent version of *DAISY* is based on a neural network approach called plastic self organizing maps (PSOM) [O'N07]. The system has been successfully applied to a variety different insect species such as bumblebees, moths, wasps, midges, butterflies, larvae, and spiders. Although the results achieved by *DAISY* are comparable to the performances of specialized systems such as *ABIS* and *SPIDA-web*, user interaction is still required to manipulate the specimen, capture the image, and segment the region of interest which ultimately hampers the throughput of these systems for large datasets.

In the recent past a considerable amount of literature has been published on insect classification to overcome the limitations of *ABIS*, *SPIDA-web*, and *DAISY*. An autonomous system for bee classification named *DrawWing*[14] was proposed by Tofilsky in [Tof08]. Unlike *ABIS*, Tofilsky's approach is not only based on standard morphometry of wing venation but also on characteristic landmark points, so called geometric morphometrics. Because points of interest are automatically located by the software based on vein junctions, no user interaction is required to align wing images. However, before image capturing the wings should be detached from insects bodies to achieve good results, which not only requires a significant amount of user interaction to prepare the specimen but also harms the animal before classification. In [LDZ$^+$07], Larios *et al.* proposed a combined hardware-software system for automatic taxonomic insect classification using histograms of local appearance features. To automatically categorize stonefly larvae to the species level, the authors developed a mechanical system for photographing the specimens under a microscope and provided a software tool for region detection, feature extraction and classification. Larios *et al.* propose a BoW approach [CDF$^+$04] where in a first step three different detectors are used to locate regions of interest. Each detected region is subsequently represented as SIFT features [Low04] and k-means clustering over all descriptor vectors is applied to build the keyword dictionaries. Codewords are then defined as the centers of the learned clusters. Thus, each detected patch of the larvae is mapped to a certain codeword through the clustering process. Each input image can therefore be represented by the histogram of codewords. The combination of three different detection algorithms boosts the performance of the algorithm significantly and outperforms systems solely based on only one of the used detectors, achieving accuracies of up to 82% for a four-class problem. A similar approach named *BugID* was used by Lytle *et al.* [LMMZ$^+$10] for automatic classification of benthic invertebrate samples. Similar to the approach by [LDZ$^+$07], Lytle *et al.* apply three different region detectors to localize object instances at different scales and shapes. Each detected region is represented by SIFT features [Low04] and subsequently compared to samples in the database using random forest matching approach [Bre01]. However, while previous approaches for insect classification treat the problem in a closed-set fashion, Lytle *et al.* propose to utilize an open-set classification scheme. While in closed-set classification it is assumed that all possible classes are known to the system, an open-set system first has to decide if a probe is known to the system

[13] `http://www.tumblingdice.co.uk/daisy` Last visit: January 17th, 2014
[14] `http://drawwing.org/` Last visit: January 17th, 2014

before it is actually assigned to a certain class of the training set. Such a procedure helps biologists to quickly identify novel species not yet present in the database which, according to [LMMZ+10], is the common scenario for most field-collected samples. The experiments on a dataset of 9 different species shows that 94.5% of correctly accepted samples was classified correctly while most unknown species were correctly rejected by the system. In 2012, Utasi proposed a local appearance feature based approach for automatic categorization of tarantulas in [Uta12]. While *SPIDA* focus on the external genitalia of spiders to classify the species, Utasi *et al.* followed a more general approach by using local color descriptors known as colorSIFT [BG09], a variant of SIFT applied to different color channels. After feature extraction the BoW model [CDF+04], a histogram representation of visual features, is adopted to obtain a more compact and meaningful representation. In [Uta12] the authors compare different state-of-the-art classification methods such as Naïve Bayes Classification, linear Support Vector Machines (SVM), and Supervised Latent Dirichlet Allocation (sLDA) [BNJ03], were the latter performed best with an accuracy up to 77% on a dataset of 7 different species.

Although the majority of proposed algorithms for species classification are specialized to categorize insects, there has been an increasing amount of literature on classification of larger animals such as fish, reptiles or mammals in recent years. In 2010, Rodrigues *et al.* presented an approach for automatic classification of fish species in [Rod10]. For feature extraction the authors propose to apply Principal Component Analysis (PCA) on vectorized color channels in the YUV color space to encode both brightness and color information. Next, unsupervised clustering techniques based on two immunological algorithms, Artificial Immune Network (aiNet) [dCvZ01] and Adaptive Radius Immune Network (ARIN) [BBCZ05], are utilized in order to find natural groupings of features extracted from different individuals of different species. By this means the resulting clusters refer to features gathered from individuals of the same species. Finally, a Nearest Neighbor Classifier based on the distance between a new feature vector and the obtained cluster centroids is used for classification. The proposed algorithm is evaluated on a dataset of 4 different fish species and compared with a SIFT matching approach. Although the system outperformed SIFT matching and achieved an overall accuracy of 92%, several images of only one individual per species was present in the dataset, which makes it hard to estimate the generalization capability of the system. Furthermore, the dataset was gathered in a controlled environment with constant background to eliminate variations resulting from background clutter and challenging lighting conditions which are often present in a real-world environment. Spampinato *et al.* proposed a complete system for fish detection, tracking, and classification operating in natural underwater environments in [SGD+10]. For detection and tracking the authors used an approach developed in their earlier work [SCBNF08] (see section 2.4.2). Based on the regions of interest obtained from detection and tracking the authors subsequently extract a number of texture and shape descriptors. The texture of the object is described by statistical moments of its gray-scale histogram, Gabor wavelets, and various properties of the gray-level co-occurrence matrix. Shape information on the other hand is characterized by histograms of Fourier descriptors of the object boundary obtained from the Curvature Scale Space (CSS) image proposed by [AMK99]. Principal Component Analysis (PCA) and Discriminant Analysis are finally applied for feature space transformation and classification, respectively. An average classification accuracy of 92% was achieved on a dataset of 10 different fish species gathered under realistic real-world underwater conditions which shows the applicability of the proposed algorithm.

Also the classification of mammal species has drawn the interest by a number of computer vision experts. Kouda *et al.* for instance proposed a system to reliably differentiate between raccoons and raccoon dogs, which look very similar in shape and appearance, in [KMF11]. The authors developed an

intelligent camera trap based on face detection and recognition techniques for population monitoring for these two species. Face detection is based on Histogram of Oriented Gradients (HOG) [DT05] in combination with a Support Vector Machine (SVM) for classification. However, according to Kouda *et al.* a reliable differentiation between raccoons and raccoon dogs is not feasible based on the detection confidences. Therefore, the authors train a second SVM for species classification based on features obtained from the Discrete Cosine Transform (DCT) and feature selection. Another study by Wilber *et al.* uses lightweight techniques for animal detection and classification that can help biologist to study squirrels and tortoises in the Mojave Desert using mobile devices [WSL+13]. For animal localization the authors apply the keypoint detection algorithm used in SIFT [Low04] to extract a number of sparse keypoints. Around each detected point of interest a Local Binary Pattern (LBP) [AHP06] based descriptor is extracted and a 1-class SVM is applied to classify between target objects (squirrel and tortoises) and objects that are not of interest. For species classification Gabor features are extracted from the automatically obtained regions of interest and a multi-class SVM is used to differentiate between three different squirrel species. Results on a self-established dataset shows the effectiveness of the proposed algorithm with an average recognition rate of around 78%.

Afkham *et al.* on the other hand proposed to use joint visual texture information of detected animals and their background for animal classification [ATEP08]. Therefore, an additional segmentation algorithm to extract the animal from the background is obsolete. The authors apply a method adopted from visual object categorization based on a visual word dictionary generated from Markov Random Field (MRF) descriptors [VZ03]. Furthermore, Afkham *et al.* propose to apply a joint probabilistic model in order to obtain more discriminative features. The main idea of applying joint probabilities is to capture the likelihood that different visual words appear in the neighborhood of each other which implicitly encodes the information about context and the background surrounding the object. The proposed approach achieved promising results on a publicly available dataset of $1,239$ images of 13 animal species. A particularly interesting and sophisticated approach for automatic animal categorization of wildlife pictures captured by remote camera traps was developed by Yu *et al.* in [YWK+13]. The system exploits the sparse coding spatial pyramid matching (ScSPM) paradigm proposed in [YYGH09], a method for general object categorization. The algorithm first extracts dense SIFT features and LBP descriptors on a manually cropped image region. The weighted sparse coding scheme for dictionary learning, originally proposed by Wang *et al.* in [WYY+10], is utilized to generate a compact dictionary that can sparsely represent the incoming descriptors with minimum error. Subsequently, spatial pyramid matching, an extension of the BoW approach, is used to model the spatial layout of local image features at multiple scales. A linear SVM is finally used for classification. On a challenging self-established dataset of $7,000$ images of 18 species gathered under natural conditions from autonomous camera traps, remarkable results could be achieved by the system with an average recognition rate of 82%. However, although a variety of algorithms for automatic detection of animals in video footage exist (see section 2.4.2), manual segmentation of the animal is required in the proposed framework.

2.4.4 Face Recognition

Responsible partner/ Author: FHG / Alexander Loos
Related Technology Enablers: TE-204

Automatic face recognition is one of the most challenging problems in the field of computer vision and image understanding. It has therefore been one of the most successful applications of image

analysis and received significant attention by many researchers and computer scientists over the past three decades. The reasons for this trend are twofold: First, feasible technologies for automatic face recognition in images or videos have a wide range of many commercial and law enforcement applications such as advertising, market research and surveillance and are therefore also important in the context of selected MICO showcases. Second, after more than 30 years of research, significant progress has been made in the field of automatic face recognition and biometric identification and algorithms for robust and accurate identification are nowadays commercially available. Table 5 gives an overview of some commercial face recognition systems.

Commercial Product	**Company**	**Website**
FaceVACS	Cognitec	`http://www.cognitec.com/`
SEKUFace	EUROTECH	`http://www.eurotech.com/en/products/SekuFACE`
FaceExaminer	MorphoTrust	`http://www.morphotrust.com/Technology/FaceRecognition.aspx`
IWS Biometric Engine	ImageWare Systems	`http://www.iwsinc.com/`
BioID	BioID AG	`https://www.bioid.com/`
Visual Casino 6	Biometrica	`http://biometrica.com/`
MFlow Journey	Human Recognition Systems	`http://www.hrsid.com/company/technology/face-recognition`
Picasa	Google	`http://picasa.google.de/intl/de/`
IPhoto	Apple Inc.	`http://www.apple.com/mac/iphoto/`

Table 5: Available commercial face recognition systems for surveillance and entertainment. Note that some of the links might have changed.

Completely autonomous face identification systems usually consist out of three main parts: face detection, face alignment, and face recognition. In the field of face and object detection, plenty of research work has been carried out during the past ten years, see also [Gar04]. In general, approaches can be distinguished into attention-based, appearance-based, feature-based and rule-based methods for object and face detection. Object-based methods use color or motion information, and hence cannot be used in grey value images and still images. Appearance-based methods use neural networks and work directly on grey value images, providing exact results and high recognition rates, while being computationally expensive. Feature-based approaches combine high detection rates and efficient processing, but have the drawback of high false-positive rates. Finally, rule-based approaches, which use heuristics like geometric knowledge about occurring curves within the face, are efficient but have comparatively low recognition rates. One of the best-known face detection approaches is the one from Viola and Jones [VJ01], which has been implemented as part of the publicly available OpenCV-Library[15]. The approach uses boosting for training and Haar-wavelet like features. Its drawback is the use of relatively complex features which are computationally expensive. A similar approach is [KE06], which includes improvements regarding feature extraction, grid search and feature combination, using less complex features in a first classification phase, and more complex features in a second classification phase. One out of three feature types can be applied: edge orientation features, census features [ZW94], also known as local binary patterns [OPH96], or scaled versions of census features. By considering only local intensity differences, the features are robust against illumination changes. Each classifier consists of

[15] `http://opencv.org/` Last visit: April 25^{th}, 2014

look-up tables which are built up during offline training using Real-AdaBoost [SS99]. For a complete review of face detection algorithms the reader is referred to [ZZ10, Gar04]

Face recognition applications itself are mainly used for three tasks:

1. Verification (one-to-one matching): The face recognition system has to determine if the person on the image is who he/she claims to be.

2. Identification (one-to-many matching): Out of a limited set classes, the face recognition system has to determine the identity of the person of the image.

3. Watch-List: In an open-set classification scheme the face recognition system first has to decide if a person is known or unknown, i.e. is part of the training set. If not, it has to reject the person as impostor. If however the person is known to the system it has determine his/her identity.

There are a number of challenges in automatic visual facial analysis and face recognition in particular. First, although rigid object detectors are generally used to localize faces in images and videos the human face generally is a highly deformable object. Different facial expressions for instance can vary the visual appearance of a face significantly. Moreover, numerous other extrinsic and intrinsic factors may cause the appearance of a face to vary [GMP00]. Intrinsic factors are purely influenced by the physical nature of the face such as age and different facial expressions. Extrinsic factors on the other hand arise from outside of the individual. These factors include illumination, pose, occlusion, scale and imaging parameters such as resolution, focus or noise among others. Thus, a facial recognition system should be robust against those kinds of image variation and a dataset for thorough evaluation should include such factors. However, face recognition algorithms are often evaluated on official benchmark datasets gathered under laboratory conditions where they usually perform quite well. Once such techniques are applied in real-world environments often a significant decrease in performance can be observed. However, recently published *Labeled Faces in the Wild* (LFW) dataset [HRBLM07][16] provides a challenging benchmark dataset to test face recognition approaches in unconstrained conditions. An important part of face recognition research is a thorough evaluation and benchmarking of developed algorithms including a detailed reporting of the experimental setup. The Face Vendor Recognition Test (FRVT)[17] for instance provides independent evaluations of commercial and academic face recognition algorithms under challenging and realistic conditions using standard performance measures. This not only helps the face recognition community to identify future research directions but is also an opportunity for researchers to easily evaluate reported results and benchmark their systems against state-of-the-art algorithms as results become independently reproducible.

Machine vision for automatic face recognition has not only attracted interest of computer scientists but from diverse scientific disciplines such as image processing, pattern recognition, computer vision, machine learning, computer graphics, and psychology. This and because of the vast and diverse amount of literature published in the field of automatic face recognition makes it difficult to find a generic taxonomy of face recognition algorithms. Moreover, a complete literature survey of existing face recognition techniques is out of the scope of this report. For a more complete and detailed overview of the state of the art in automatic machine vision approaches to identify humans by their face the interested reader is referred to [AAA+11, CWS95, ZCP03, JA09, TEBEH06, Vij13].

[16] http://vis-www.cs.umass.edu/lfw/ Last visit: April 22nd, 2014
[17] http://www.nist.gov/itl/iad/ig/frvt-home.cfm Last visit: April 15th, 2014

According to [JA09] face recognition techniques can broadly be divided into three main categories based on their image acquisition protocol:

1. Algorithms that utilize data from multiple sensors, e.g. stereo cameras, infrared cameras or 3D sensors.

2. Methods that operate on intensity images obtained from a single camera.

3. Techniques that perform face recognition in videos.

In this report we concentrate on the second category which is by far the most applicable one in real-world non-intrusive situations. Furthermore, for face recognition in video often the techniques developed for still images are applied to a few selected frames after face detection and tracking [CWS95]. However, recently approaches were proposed which incorporate multiple modalities in order to increase the performance of a face recognition system. Steffens *et al.* [SEN98] utilize stereo information to increase the robustness of the system while [CCJP99] exploits visual and audio information.

Early approaches developed in the early and mid-1970s measured facial attributes such as the distances between certain facial landmark points and used these as unique features for identification [Kan73, Kel70]. However, a precise automatic localization of facial landmarks is hard to achieve in practical applications. Moreover, simple distance measures are usually not robust to the intrinsic and extrinsic factors explained above. However, with the progress on statistical and machine learning techniques in the early 1990s, new interest in automatic face recognition approaches arose. Zhao *et al.* devide face recognition methods for intensity images into three main categories [ZCP03]: *Holistic Appearance-based Approaches*, *Local Keypoint Methods*, and *Hybrid Techniques*.

Holistic Appearance-based Approaches Holistic approaches are of the most successful and very well studied techniques for human face recognition. They use the whole face image as input to the recognition system. Many holistic techniques even use the raw gray scale pixel intensities as features. These methods are often based on subspace methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Locality Preserving Projections (LPP) or even random projections. Here, the high dimensional vectorized face images of size are usually projected into a lower dimensional subspace of size using a unitary projection matrix. The resulting low-dimensional feature vectors can then be used for classification. Typical implementations of holistic appearance-based approaches are Eigenfaces (PCA) [TP91b, TP91a], Fisherfaces (LDA) [BHK97], Laplacianfaces (LPP) [HYH$^+$05, He05] , Randomfaces [WYG$^+$09]. These methods are solely based and simple gray-scale information. However, a lot of effort has been put into the development of compact yet discriminative feature representations (see section 2.4.1). These descriptors are also used in the context of face recognition. Gabor features for instance are known to perform well in face and pattern recognition tasks for humans. Recently, Gabor features have been used in combination with the Sparse Representation Classification (SRC) scheme by [YZ10] and outperformed the original Randomfaces algorithm by [WYG$^+$09], which uses basic pixel information as features. Furthermore, Ekenel *et al.* propose to represent the appearance of human facial regions using the Discrete Cosine Transform (DCT) on non-overlapping blocks of the aligned face [ES05, Eke09, SES07]. Ahonen *et al.* were the first who applied LBP to the field of face recognition [AH04, AHP06]. Like in the BDCT approach by [ES05] the face image is first divided into equally sized non-overlapping blocks. A LBP histogram is extracted for each region and the final feature vector is formed by concatenating the resulting histogram sequences. Ahonen *et al.* propose to use a simple χ^2-distance based nearest neighbor approach

for classification. One known disadvantage of LBP however is that it may not work well on noisy images or flat regions such as cheeks or the forehead of human faces due to its thresholding paradigm which is solely based on the gray level value of the center pixel. Moreover, the reliability of LBP in known to decrease significantly for large illumination changes and shadowing. To overcome these limitations, Tan and Triggs proposed to replace LBP with a three-level operator called Local Ternary Patterns (LTP) in [TT10]. In 2005, Zhang *et al.* successfully combined Gabor wavelets and Local Binary Patterns to form a new face descriptor called Local Gabor Binary Pattern Histogram Sequence (LGBPHS) [ZSG+05]. First, so called Gabor Magnitude Pictures (GMPs) are obtained by convolving Gabor wavelets of different rotations and scales with the gray-scale input image. Each GMP is then divided into equally sized non-overlapping regions from where LBP histograms are extracted and concatenated to form the final face representation. For recognition, histogram intersection is used to measure the similarity of concatenated histograms and the nearest neighbor classification paradigm is applied to obtain a final decision. Experimental evaluations on publicly available benchmark datasets showed the effectiveness and robustness of the proposed approach.

Although some of the above mentioned methods try to incorporate local information by applying features that efficiently describe the region within a local neighborhood or by dividing the facial image into non-overlapping blocks, the outcome of each of theses approaches one single feature vector that is a representation of the global appearance of the face. However, recently also other techniques were proposed and evaluated that explicitly exploit local information. The most important and promising ones are briefly reviewed in the subsequent section.

Local Keypoint Methods In contrast to holistic methods, face recognition techniques based on local keypoints first try to detect distinctive facial landmarks such as eyes, nose, and mouth and then measure the geometric relationship between those keypoints. Standard statistical pattern recognition algorithms are subsequently employed to match extracted geometrical features. Early approaches in the field of automatic face recognition were often based on these techniques. One of the earliest works in this field date back to the mid-1970s [Kan73], where relationships such as distances and angles between 16 different markings were used for recognition. More recently, Cox *et al.* reported a recognition rate of 95% on a dataset of 685 individuals. They manually annotated 35 different facial landmarks and computed a 30-dimensional feature vector based on a mixture of distances is obtained from these points. However, facial landmarks were still annotated manually. Therefore, a significant decrease in accuracy can be expected for a completely autonomous system where the location of automatically detected keypoints is consequently not as accurate.

However, with the recent success of local keypoint detectors robust powerful algorithms based on local features were proposed. One of the most well-known methods is the Elastic Bunch Graph Matching (EBGM) approach by Wiskott *et al.* in [WFKvdM97] which is based on Dynamic Link Structures [LVB+93]. In EBGM, faces are represented as labeled graphs where the nodes represent local textures obtained by Gabor features (so called "jets") and the edges represent the distances between nodes. Hence, a face is represented as a collection of keypoints and their spatial arrangement. All instances of frontal faces in the database are represented with the same kind of graph. A bunch graph is then created by combining the graphs of all faces in the database. Hence, a certain node of a bunch graph represents the texture of all variants of a specific facial landmark and the edges represent the mean distance between two keypoints. More generally, a bunch graph is an abstract representation of object classes rather than of instances of a certain object. Thus, EBGM takes advantage of combinatorics of facial landmarks to represent a new face that was not seen before by the

system. Recognition can then simple been done by comparing the graph of the new face to all graphs in the database and take the one with the highest similarity score. Albeit the fact that EBGM is one of the best performing algorithms for face recognition, it does suffer from the serious drawback of extensive ground-truth annotation. According to [Suk00] EBGM only becomes adequately dependable after manually placing the graphs for at least 70 training images. Moreover, due to the complex 3D structure of a human face the automatic placement of keypoints becomes harder for off-frontal face images. However, a considerable amount of literature have proposed techniques to recognize faces from their profiles [HKLR81, HRR78, KB76, LL99b, LL99a]. Another major drawback of EBGM is that it might not work well for data gathered from surveillance cameras due to the low-resolution character of the images and video sequences [JA09].

Hybrid Techniques It is well known from psychophysics and neuroscience that both holistic and local information are crucial for perception and recognition of faces. Thus, also a machine vision system should utilize both. One of the first hybrid approaches called *modular eigenfaces* was presented by Pentland *et al.* in [PMS94] where the authors extended their earlier system [TP91b] towards eigenfeatures such as eigeneyes, eigenmouth, etc. Experiments in [PMS94] indicate that eigenfeatures extracted on different regions of the face are much more robust against different facial expressions than the holistic eigenfaces approach of [TP91b]. This supports the claim that locally extracted features are well suited for images with large variations. Another interesting biological inspired approach for hybrid face recognition called Local Feature Analysis (LFA) in combination with PCA was presented by Penev and Atick in [PA96]. Unlike the global eigenmodes, LFA gives a description of the face in terms of statistically derived local features and their positions. In [PA96] the authors successfully combine the global face representation extracted by PCA and the local information by LFA to enhance the recognition performance of both modalities alone.

Also flexible models such as Active Shape Models (ASMs) and Active Appearance Models (AAMs) which use both shape and gray-level information have been used to recognize faces [LCC95, ETC98]. ASMS and AAMs are statistical models generic objects that are deformable so they can fit themselves to the shape of an object in a new image. After the flexible appearance model was fitted to a new face, shape parameters as well as local gray-value information at each model points are collected. Then, the face image is transformed to a mean face shape and shape-free model parameters can be obtained. All three parameter-sets, i.e. shape parameters, local gray-value information at the model points, and the shape-free model parameters are used for classification.

A pose and illumination invariant face recognition which combines 3D morphable models and component-based face recognition techniques was presented by Huang *et al.* in [HHB03]. First, a 3D morphable model of a face of every person in the database is constructed based on three face images in different poses (frontal, semi-profile, and profile). Once this model is constructed it can be used to generate arbitrary synthetic images of the same person in various poses and different lighting conditions. Then, a component-based face recognition system can be used to identify unseen test images. Similar to EBGM, the main idea of component-based methods is to decompose a face into its main components, e.g. eyes, mouth, and nose, and model the interconnections between them with a flexible geometrical model. However, in Huang *et al.* simple gray-scale components were used instead of Gabor features as in [WFKvdM97]. Although proposed system achieved impressive results in experiments conducted in [HHB03], one major drawback of the system is that the generation of the 3D model is person-specific and therefore requires cooperation in order to get high-quality images. However, recent success in face recognition based on 3D morphable models might lead to powerful and

robust face recognition algorithms applicable in real-world environments [HV13, MLB+13]

A lot of effort has also been put into face recognition using neural networks. One of the first applications of neural networks to the field of face recognition was presented by Lin *et al.* in [LKL97] where they proposed a probabilistic decision-based neural network (PDBNN) for identification. The system was evaluated on two public benchmark datasets and achieved state-of-the-art results at that time. Later a radial basis function (RBF) neural classifier was used by [EWLH02] to cope with the problem of small training sets. A hybrid learning approach was proposed to train the RBF neural network. Evaluation on publicly available datasets demonstrated the efficiency of the proposed learning algorithm with regard to classification and learning efficiency. With the recent development and success of deep learning [Hin07, HOT06, Hin09], artificial neural networks regained attraction for face recognition. Most recently, Taigman *et al.* [TYRW14] developed an algorithm called *DeepFace* which successfully combines 3D modeling of a human face for alignment and a nine-layer deep neural network for recognition. The system was trained on an extremely large dataset from *Facebook* consisting of four million facial images belonging to more than $4,000$ individuals. It was then tested on the Labeled Faces in the Wild (LFW) dataset which is one of the most widely acknowledged benchmark dataset for face verification in unconstrained environments, achieving an accuracy of 97.35% which is close to human-level performance.

2.4.5 A/V Error Detection and Quality Assessment

Responsible partner/ Author: FHG / Ronny Paduschek
Related Technology Enablers: TE-205

Audio and visual (a/v) error detection and quality assessment (QA) has a high relevance to the MICO - IO10 News showcase as the idea is to filter and rank videos based on a/v quality aspects.
Quality assessment in digital media mostly pursues a qualitative estimation of a/v footage, separately [YRH+10]. For a/v QA there are different models to assess the quality in order to supervise and finally to ensure the a/v quality. These models can be applied to the detection of errors in a/v footage as the occurrence of errors affects the QA.

Reference Models Reference models depend on the presence of a reference signal, which belongs to the original a/v footage on the supposition that the original is accurate and not compressed. Finally, one can differentiate between three different reference models: The *full reference model* needs a reference signal which is compared with the resulting signal. While the transcoding step a reference signal passes encoder and decoder. In a second step the decoded test signal is compared with the test signal in order to indicate possible quality variations between both signals. The *reduced reference model* uses signal-descriptive features in terms of a fingerprint, instead of the full reference signal. However, in many cases a reference signal is not available (*no-reference* case), i.e., there is no information about the original signal. In such cases, the quality of a signal can be measured using the signal only and under consideration of additional information about individual error characteristics [Tra01].

Video Quality Metrics Further on, one can differentiate between objective and subjective video quality (VQ) metrics. However, only objective metrics are relevant and considered to the MICO topics while in this section the subjective metrics are only mentioned for the sake of completeness. The *Video*

Quality Expert Group (VQEG)[18] deals with both in order to define VQ standards and to develop new VQ measurement approaches. VQEG describes two methods of subjective quality measurement (SSCQE[19] [KH08], SAMVIQ[20] [Tho12]). Thereby, the video data is evaluated by test persons without the existence of any reference videos. SAMVIQ is a standardized approach and convenient for comparing the individual formats, codec, and bit rate of videos. ITU (International Telecommunication Union) Recommendation BT 500-11 or the technical review of the European Broadcast Union (EBU) [KSWP05] provide further information about standardized subjective quality assessment. Objective video QA models, like Mean Squared Error (MSE) [TLT+13] and Peak Signal to Noise Ratio (PSNR) [NK13] are often used for reference-based measurement. In the past objective VQ measurement techniques were developed considering characteristics of the human visual system (HVS) [WBL02]. Structural SIMilarity (SSIM) is another reference-based measure. It is used to compare local patterns of pixel intensities in order to estimate the similarity between two images [WBSS04] [RW10]. The *Laboratory for Image & Video Engineering (LIVE)*[21], practices research in the field of QA with the focus on full an no-reference based analysis [SB12]. LIVE provides a large image and video QA benchmark set, which is used in many publications [SSBC10] [RW10] [CSRK11]. The LIVE VQ database contains ten uncompressed videos as references and 150 distorted (e.g. using MPEG-2 and H.264 compression) videos created from the reference videos.

Error Detection Approaches While QA basically measures the signal's quality with a single value, error detection methods are intended to provide statements about the existence of artifacts and their positions in the signal. Error detection in turn can be applied for QA. Digital video formats allow the detection of errors in different ways: container-based (wrapped) by checking the integrity of the container, data stream-based (coded), or base-band (decoded, on pixel or sample level) analysis. Error detection for stream-based analysis is performed after available information is picked out from the header. Base-band analysis is performed on decoded video frames considering intra-frame or inter-frame processing. Base-band approaches are applicable to detect video encoder artifacts, such as blocking, ringing, or blurring. Furthermore, visual errors can be located precisely up to a single pixel and intensities define their appearance. A variety of audio-visual errors on container-based, stream-based, and signal-based level are collected and discussed within a QC working group (Strategic Programme on Quality Control[22]) of the EBU. Due to the termed benefits base-band analysis has a major importance in the field of VQ assessment.
In contrast to the upper mentioned reference-based VQ metrics, there is a lot of work done using no-reference approaches [LJCM13] [FM05] [HTG09] in order to detect errors in a/v content. Furthermore, no-reference base-band analysis preferably need available information about the individual characteristics, occurrences and causes of errors. Major digital video standards, e.g. H.261, MPEG-1, MPEG-2 / H.262, H.263, MPEG-4 part 2, H.264/MPEG-4 AVC, etc., rely on linear block transforms such as the discrete cosine transform (DCT). These video codec formats are based on transform blocks (macroblocks), with 8x8, 16x16, or any different sizes[23] of luma samples. Early standards used motion compensation relating on entire macroblocks, the transform block size has always been 8x8. Newer formats provide enhanced prediction accuracy [STL04].
Consequently, the most common coding artifact is blocking, which occurs particularly at lower bit

[18] `http://www.its.bldrdoc.gov/vqeg/` Last visit: February 19th, 2014
[19] Single Stimulus Continuous Quality Evaluation
[20] Subjective assessment methods for video quality
[21] `http://live.ece.utexas.edu/research/Quality/index.htm` Last visit: February 20th, 2014
[22] `https://tech.ebu.ch/groups/qc` Last visit: March 26th, 2014
[23] Prediction partitions can have seven different sizes, as 4x4, 8x8, 16x16, 8x4, 4x8, 8x16, and 16x8

rates [Vla00]. Blocking artifacts in images or videos are detected in different ways [Vla00] [YWK10] [UF11]. Methods to detect further coding artifacts, as ringing and blurring are described in [BS05] [LKH10] [LJCM13]. Coding errors as mentioned above are also applicable to be detected in images as the information of only a single frame can be sufficient for their detection. In contrast and due to their physical characteristics, the detection of ”temporal” errors like freezes [HTG09], (de-)interlace artifacts [FC09], or field order correctness [Bay07] mostly need information of at least two consecutive frames.
No-reference-based approaches result in measurement values which in most cases can directly be used for QA if they are normalized in order to provide a comprehensible quality measure. Furthermore, QA can be stabilized combining results of several error types assumed that specific errors only occur in combination or not at the same time. Same can be applied when container information is cross-checked with audio and/or video error detection results.

2.4.6 Temporal Video Segmentation

Responsible partner/ Author: FHG / Ronny Paduschek
Related Technology Enablers: TE-206

The increasing amount of multimedia information leads to a greater need of efficient ways to retrieve and to search the content of interest. Temporal video segmentation (TVS) can considered as a first step towards automatic annotation of digital video for browsing and retrieval [KC01] in order to automatically extract the structure of videos. TVS comprises a lot of video analysis methods for temporal video feature extraction, e.g. keyframe extraction, shot similarity detection, or shot structure analysis in order to analyze the frequency of consecutive shot boundaries. All these methods can be used for automatic video annotation, browsing, and retrieval. TVS also has a high relevance in the field of dramatic composition of feature films [RS03],[Lie01]. The Text REtrieval Conference Video Retrieval Evaluation (TRECVid[24]) encourages research in information retrieval and provides a large collection of test data, a uniform scoring, and the opportunity of comparing their results [SOK06]. TRECVid also provides amongst others a shot boundary detection tool for evaluation issues and comparison to other work in this domain. Following, the most common sub-domains of temporal video segmentation are introduced.

Video shot detection Commencing with the extraction of temporal information by detecting video shot[25] boundaries, further temporal and non-temporal video analysis is able to be applied on the enclosed video segment, defined by the detected video shot boundaries. Video shot detection provides basis for existing video segmentation methods and can be considered as initial step for realizing further video analysis tasks [Ham09]. Since video shot detection is a very meaningful task in the TVS domain, a lot of work following different approaches, e.g. pixel-based, feature-based, histogram-based, statistics-based, transform-based, motion-based methods, and combinations of them [NMF+98],[CLHA08],[Ham09] has been done. A survey on shot boundary detection on the related work of the annual TRECVid can be found in [SOD10]. [CGP05] gives a concisely overview of features and methods for shot boundary detection.
Since video filters or cut effects are used in common post-production processes and thus characterize the composition of video segments, automatic video shot detection followed by the detection of gradual transitions is a very meaningful tool, especially for the feature film domain.

[24] http://trecvid.nist.gov
[25] A shot can be regarded as a continuous sequence of video frames without any interruption and recorded by a single camera.

Gradual Transition Detection Similar to shot boundary detection transitions are detected by comparing two consecutive video frames by a distance function in order to compare the similarity between a frame pair followed by defining thresholds to make a decision [BCM+05]. Further work deals with training processes followed by a classification step, using neural networks [LZ01], or a probabilistic based algorithm [Han02]. Progressive or gradual transitions can be separated into different types, e.g. dissolves, fades, and wipes. In addition to that a lot of research has been done using color histogram-based, standard deviation-based, and feature-based methods [ZMM99],[Lie01],[GC07] in order to detect transition types in videos.

Keyframe Extraction A keyframe can be considered as a selected video frame that represents a video segment depending on its content and with a high visual semantic importance. A simple keyframe can be the first, last or an arbitrary frame of a shot, however without the consideration of visual semantics. The size of a keyframe set can be fixed as a known *priori*, left as an unknown *posteriori*, estimated by the level of visual change, or *determined* in order to find the appropriate number of keyframes before the full extraction is executed [TV07]. In the literature automated detection of keyframes is described in different ways: [SA07] uses MPEG-7 color descriptors [OCK+03] and texture features locally extracted from keyframe regions. Each frame is described as in terms of higher semantic features using a hierarchical clustering approach. [SG10] and [HCL04] introduce methods in the compressed video domain. Further work on keyframe extraction methods is summarized in citeHXL:2011.

Shot similarity and Scene Detection The scene detection is based on the fact that video shots belonging to the same scene have high visual semantic relations in order to be grouped into a high-level story unit (scene). Since a scene is a complex and complicated concept information about film dramaturgy and film-production techniques would be helpful in order to adapt these clues to a scene detection algorithm.
Scene detection is handled differently in previous research work: [RS03] uses a keyframe comparison method considering a similarity measure of a given shot with respect to previous shots. [TZ04] deals with clustered shots based on background similarity by extracting visual features from selected regions of keyframes. [TVD03] uses multi-resolution edge detection followed by a neighborhood visual coherence measure at shot boundaries which in turn are estimated by taking similarity of colors into account. A broad survey on scene segmentation and other important issues to video analysis and video retrieval can be found in [HXL+11].

Temporal video segmentation is useful for the MICO - IO10 News showcase for validation, annotation, and editing of (news) video segments. Keyframe detection can be used to provide thumbnails of selected video segments for easier navigation.

2.4.7 Speech-Music Discrimination

Responsible partner / Author: FHG / Jakob Abeßer
Related Technology Enablers: TE-207

The problem of automatically discriminating between speech and music in audio recordings was first investigated in the late nineties among others by Scheirer and Slaney [SS97]. Speech-music discrimination has many potential applications. For instance, dividing audio streams into segments of music and segments of speech allows to effectively choose algorithms for signal enhancement and

audio coding, which are usually specialized towards different characteristics of these two signal types. Also, speech recognition applications benefit from a preliminary detection of speech segments in order to avoid misclassifications. Special requirements towards real-time performance and robustness to noisy signals arose from the application of speech-music discrimination in the monitoring of telecommunication systems and broadcasting services such as radio channels.

In the literature, speech-music discrimination is usually considered as a two-class classification problem. Occasionally, some authors include additional classes such as silence or environmental sound [PT05]. Spoken voice is commonly assumed to have a limited bandwidth, strong energy fluctuations, as well as short durations of acoustic events (vowels and consonants). On the other side, music signals are modeled as mixture of harmonic (tonal) and percussive (noise-like) note events. The harmonic notes most often show constant fundamental frequency values (apart from modulation techniques such as vibrato and glissando) whereas speech commonly shows continuous fluctuations of the fundamental frequency over time.

Most of the features that have been applied for speech-music discrimination so far are low-level and mid-level features from the field of speech recognition and music information retrieval (MIR). Fu et al. propose a categorization of features into four categories [FWX09]. The first group of features such as the root mean square (RMS) or the modulation energy, relate to the dynamic properties of an audio signal and characterize the signal amplitude. The second group of features describe the short-time magnitude spectrogram of the signal using different statistical measures such as the spectral centroid, spectral flux, spectral roll-off, or zero-crossing rate [PT05]). The third group of features such as Mel Frequency Cepstral Coefficients (MFCC), which are also applied for speech recognition, Delta Cepstral Energy [CPLT99], Power Spectrum Deviation, and Modified Low Energy Ratio (MLER) [WGY03] describe the spectral envelope. The last group of features such as the Average Pitch Density (APD) [FWX09] describe the tonal characteristics and the temporal progression of the fundamental frequency. With all these features being computed on short time frames of length 10-20 ms, the pulse metric and the rhythm metric proposed by Scheirer and Slaney [SS97] and Jarina et al. [JOMM02] are computed based on long-term autocorrelation in order to recognize music segments based on characteristic (rhythmic) repetitions. For the statistical modeling of the classes, common approaches such as Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), Support Vector Machines (SVM), and Bayesian Networks are applied.

Most papers use clean audio signals for evaluating the speech-music classification. Fu el al. discussed that additional noise and channel distortions must be taken into account since they occur in real-life applications [FWX09]. In order to avoid algorithms that are specialized only to a small set of musical styles and languages, Carey et al. emphasized that it is necessary to include a wide selection of music genres and languages into the evaluation set [CPLT99]. State-of-the art publications achieve up to 97 % accuracy [PT05] on clean audio data. The classification accuracy was shown to decrease with decreasing signal-to-noise ratio, e.g., Fu et al. reported an accuracy of 81.7 % for an SNR of 5 dB. Among others, future work must address open problems such as mixed segments that include both music and voice.

2.4.8 Music Annotation

Responsible partner / Author: FHG / Jakob Abeßer
Related Technology Enablers: TE-208

From a practical perspective, only a small number of products and companies exist in the field of music metadata annotation and enrichment. All of them offer their services for international customers via internet services. Established companies like AllMusic[26] or MusicLine[27] provide conventional music metadata such as reviews, production metadata and recommendations. All data is manually entered by editors, which is costly and time-consuming for new releases. Relatively few companies offer at least partially automatic music annotation services, the most prominent probably being the US-based company EchoNest[28]. Though its open API, EchoNest offers access to their services for any developer. Another US-based company is OneLlama[29], in Europe there are BMAT[30] and MusicDNA[31]. While the aforementioned services offer a variety of products and information, they only partially cover the detailed, specific music properties intended for for detailed analysis and targeted recommendation in the project, which may include e.g. music style/genre, emotion, mood, color, texture, tempo, distortion, instrument density, instrumentation, and musical segmentation. Such topics are addressed by the comaparatively young research field of Music Information Retrieval (MIR), which has emerged over the last ten years with the quick proliferation of the internet and lossy codecs such as MP3. It quickly became inevitable that suitable techniques were needed to handle the large amounts of music content and corresponding metadata that is available online [Brandenburg2009].

2.4.9 Music Similarity

Responsible partner / Author: FHG / Jakob Abeßer
Related Technology Enablers: TE-209

Rapidly increasing multimedia archives require for fast and efficient algorithms for search, retrieval, and recommendation of music recordings. The majority of music search and retrieval strategies rely on collaborative filtering [Cel08], i.e., they draw conclusions about music data from the way consumers and experts interacted with it. The service Last.FM[32] uses collaborative filtering based on statistics about the listening behavior of millions of users. Based on these statistics, users are clustered into groups and music that one person in a group likes is recommended to the others in the same group. Obviously, newly released songs that have never been heard before cannot be recommended. The alternative to collaborative filtering is content-based search and recommendation. One prominent example is the US-based internet radio service Pandora[33], that recommends music based on manually entered music metadata. It basically recommends songs that share similar profiles. Clearly, the manual annotation of songs brings very relevant metadata, but is extremely expensive and time consuming. Aupeo[34] is a platform that uses automatic, content-based recommendation.

We want to pursue the same technology in Mico, details about the planned approach and the challenges are given in the next chapters. The project aims to develop a software application that analyzes the content of large databases of digital music files and automatically determines descriptive

[26] Retrieved online: `http://www.allmusic.com/about`
[27] Retrieved online: `http://www.musicline.de`
[28] Retrieved online: `http://the.echonest.com`
[29] Retrieved online: `http://www.onellama.com`
[30] Retrieved online: `http://www.bmat.com/company/`
[31] Retrieved online: `http://www.musicdna.com`
[32] Retrieved online: `http://www.last.fm`
[33] Retrieved online: `http://www.pandora.com`
[34] Retrieved online: `http://www.aupeo.com`

meta data about musical properties of these files. Existing services like Pandora have shown that the data we want to extract is extremely valuable for personalized music search and recommendation. However, Pandora uses manual annotation (labeling) of music files conducted by music experts. It is clear that this is prohibitively expensive for large music databases. Instead, we want to extract this kind of information automatically. The advantage is, that our application will be able to enrich the meta data of a music database in a very short time frame and it can instantly process new music files that are ingested into the database.

Figure 6 Principle approach of automatic music analysis.

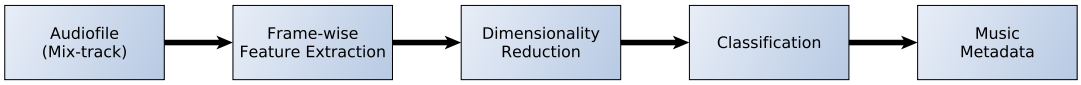


Image 6 shows the principle flowchart of our approach. Music recordings are commonly represented using a set of audio features. Suitable audio signal processing algorithms perform measurements of the audio signal properties in short time intervals (around 10-30 ms). These features are a compressed representation of the physical signal and quantify different perceptual and musical properties with regards to rhythm & tempo, tonality & harmony, or timbre & instrumentation [BSW+11].

In the literature, a multitude of audio features have been proposed that can be used for this kind of analysis. The most well-known ones originate from speech processing and are called Mel-Frequency Cepstral Coefficients (MFCC [Sla94]). They are one example for so-called low-level features. If they are further processed by statistical measurements or musically motivated post-processing schemes, one can derive so-called mid-level features (e.g., TRAP [HS99]).

The notion of musical similarity is interpreted as the distance of two recordings within the feature space. The similarity between two music pieces can be measured by computing the distance between feature vectors e.g. using the Euclidean distance or the Earth mover's distance. Another approach is to model feature vectors from the same song as probability distribution in the feature space using for instance Gaussian Mixture Models (GMM) and measure the distance between these distributions, e.g. with the Kullback-Leibler divergence [ME05].

High-level properties such as instrumentation, music genre, or mood are often derived by learning statistical models based on given training data sets (e.g., for guitar, violin, and piano) [JER09]. These models allow to extract annotations not only for whole music pieces but also for segments of music pieces. Hence, we can derive results such as "piano plays from second 0:00 to 0:30" and can store this as a high-level feature in a database. In this supervised learning scenario, the challenge with these approaches is that usually the low-and mid-level features are not directly linked to the desired high-level attributes. This is often termed the "semantic gap" and usually becomes evident in low scores for the achievable classification accuracy. One typical risk is that the machine learning components are used in such a way that they are specifically tuned to the annotated data (so-called training set or ground-truth) and achieve good recognition scores on these data. For real-world data, however, they fail completely. It takes experienced researchers to circumvent these situations.

2.5 Conclusions

The state of the art considerations of cross-media analysis quite naturally becomes a matter of the state of the art in a number of promising, but more specific, types of analysis. For the broad case general graph methods of Section 2.1.1 may be necessary, but for analyzing natural language much deeper and more nuanced (and by necessity then more narrowly applicable methods) must be applied, as discussed in Section 2.2. Finally, while flexibility of representation is important, the weight of authorship for complex extractors cannot be expected to be passed onto non-experts, and as such the more restrictive, but very hands-off interactive learning are an important addition. Notice, however, that this to some extent forms a hierarchy, general graph systems can deal with text and trees, interactive learning can say some things about text, and so on. A lot of potential power lies in the mixing and matching of tools and techniques.

Such potential for combining approaches exists especially when textual and audio-visual extractors are combined. For instance, as stated in section 2.4.1, the extraction of robust and descriptive visual low-level features is a prerequisite for accurate detection and recognition of objects, animals, and faces in particular (see section 2.4.2, 2.4.3, and 2.4.4). However, analysis of multimedia content which is solely based on low-level visual feature extraction is quickly stretched to its limits. Incorporating multiple modalities in form of audio, video, and text is crucial for more accurate and robust results. Hence, MICO will focus on the development of such cross-model approaches that fuse multiple modalities.

3 Metadata Publishing

Responsible partner / Authors: UP / Florian Stegmaier, Kai Schlegel, Emanuel Berndl
Related Technology Enablers: -

D3.1.1 Metadata Publishing describes the state-of-the-art on multimedia modelling, its analysis, preparation, processing, and publication in order to lay down a well-grounded baseline for the other processes of the project. The input consists of the raw data delivered by the extractor procedures of preceding steps. Therefore, this chapter covers the following topics:

- Multimedia Modelling, which starts by depicting ways to fragment multimedia items, letting the user specify information more accurately to the appropriate part of the item, and then leading to nowadays standards for the annotation of multimedia.
- Semantic Web-aware Description of APIs, describing common ways to embed the retrieved annotations in a Semantic Web aware format. This process is also supported by RESTFul services.
- Trust, defining the whole chain of modelling trust in different steps, namely representation, trust and distrust, metrics, propagation, and aggregation. Trust finds its field of application in MICO by enhancing the extraction step and its corresponding results as well as recommendation processes that will be depicted in WP5.
- Provenance, which is similar to trust in its use and application, but focuses on the origin and modification of utilised files. There are also frameworks discussed in this section, which show how provenance features are modelled.

As the modelling of metadata and annotations as well as information about trust and provenance is a concern for every MICO use case, this chapter will not specifically focus on single use cases, but emphasises a universally useable model for the whole project that can be adapted accordingly. Additionally to this fact, using common and well-known semantic web standards gives the possibility of opening the data that we extract and produce for linked data purposes outside the MICO context.

Throughout all of these topics, ontologies of the semantic web, and frameworks that we will use here, an important point to mention is the Resource Description Framework RDF[35], which supports the base vocabulary for our model and the aforementioned technologies. RDF encodes data in the form of subject, predicate and object triples. The subject and object of a triple are both URIs that each identify a resource, or an URI and a string literal respectively. The predicate specifies how the subject and object are related, and is also represented by a URI.

3.1 Multimedia Modelling

The human perception as well as the way content of media resources can be interpreted leads to imprecision and subjectivity. Every human, depending on his social or cultural background, has his or her own perception with respect to the actual meaning of a media resources content. Along with this, only the context defines the semantically meaningful parts of a media resource. In this light, Smeulders et al. [SWS+00b] defined (among others) two gaps that are present in all facets of multimedia information retrieval. The first to mention is the sensory gap:

[35] http://www.w3.org/RDF/

> The sensory gap is the gap between the object in the world and the information in a (computational) description derived from a recording of that scene.

With respect to the topics covered by this report, the sensory gap is not in central focus. In contrast to that, the semantic gap is omnipresent:

> The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.

The problem arising from the semantic gap is also depicted in the MICO Description of Work, and the course of the project might produce outcomes that can help to solve it. Another big point that is to be mentioned here, is the level of features that can be attached to a given multimedia item, which are inherently present in the course of consuming multimedia. Namely, these are known as **high-level features** and **low-level features**. The latter can (mostly) be extracted and defined by computers in an automated fashion, and they describe particular characteristics and signatures like color, shapes, edges or texture. These are used in a broad variety of strategies, but they also have different domains where they are best fit to be used. Popular examples (from the MPEG-7 standard) are: Dominant Color, Scalable Color, Color Structure, Color Layout, and Edge Histogram. The main difference between the two feature levels is that low-level features do not contain semantic information. By involving interpretations of the given multimedia item, meanings, semantics, and conclusions of the item can also be inferred and utilised. This is something that can generally not be done automatically and accurately by a machine, human interpretations are still needed in most cases. These fall into the domain of high-level features, and annotating is a common way to achieve semantically enriched multimedia items. In order to do this, the items can first be fragmented (see section 3.1.1), to achieve more refined process, and then annotated accordingly (see section 3.1.2).

3.1.1 Fragmentation of Multimedia Items

The fragmentation of multimedia is common in the computational era of today. It has multiple advantages when consuming the items, for example instead of describing the whole video, you can only address certain fragments of the video. As a consequence, the description gets more dignified. Just like the description, referencing subparts of videos or images is also possible and has never been easier. There are several standards and approaches in terms of fragmentation of multimedia items. A very good overview is given by [LWO+12]. Non-URI based methods exist, like MPEG-7 [Mar04b] or the Synchronized Multimedia Integration Language SMIL[36]. Besides the inability to support the fragment as a URI, the spatial and temporal information of the fragment is divided up into several different attributes. Both of these points speak against the utilisation of these standards in our context, as not supporting the fragment with a single URI does not quite fit into the idea of Linked Data as well. MPEG-21 [BPVdWK06] has got some improvements compared to the MPEG-7 standard concerning the URI and the syntax, but seems to be over the top and to complex to fit.

The Multimedia Metadata Ontology [SS10] solves some of these shortcomings by supporting a semantic description of rich, structured multimedia content. By doing so, the management, archival, retrieval, and reusability of the content is increased tremendously. Also, the different (sometimes only single) multimedia formats that are supported by the previous standards can be combined and reconciled.

[36] http://www.w3.org/TR/smil/

The most promising approach is posed by the World Wide Web Consortium in form of the Media Fragment URI 1.0[37], which itself is part of the Ontology for Media Resources 1.0 discussed in section 3.1.2. Here, the multimedia fragments are constituted by the URI of the original resource combined with an addition for a URI fragment (depicted by a # character) or a URI query (depicted by a ? character). An example is the URL `http://www.random.com/randomvideo.ogv#t=10,30`, which leads to a fragment only showing the seconds from ten through thirty of the video *randomvideo.ogv*. The difference between a query and a fragment is that while the former produces a new resource, the latter will provide a secondary resource that is related to its primary item. There are four dimensions of fragmentation supported by this standard. All of them are implemented by adding a key/value pair of the specific dimension to the URL. Temporal features restrict the time interval that is shown to given boundaries. The interval is half open, which means that the starting point of the given interval is always included, while the endpoint is excluded. Different time zones as well as time modes can also be utilised. Spatial fragmentation restricts the visible screen area to a given rectangle in relation to the format of the original multimedia object. The rectangle is specified by its x and y coordinates of its upper left corner in addition to its height and width. All of these values can be declared as an amount of pixels or a percentage value. Next to these two basic dimensions, the track and id dimensions are only supported in the advanced version of the Media Fragments[38]. If there are multiple tracks supported in the original media (for example english and german), those can be chosen via the track key, while the id gives opportunity to select various named fragments if supported (for example chapters or scenes). All of the dimensions can be combined using the & operator between the given key/value pairs. For an exhaustive listing of the possible pairs, we refer to the aforementioned standardisation respectively.

3.1.2 Annotation of Multimedia Items

For the multimedia content of today, a plethora and diversity of standards and formats exist. By enriching these multimedia items with metadata, it has become more and more feasible to manage and deal with these files, but still there is a need of finding a mechanism to combine and channel all of this information into one commonly understandable format or interface. This circumstance also applies for our extractor step, that was depicted in WP2, as the input as well as the output holds a plurality of different formats.

The process of annotating items in the internet as well as annotations themselves are also very common and widely used in the web of today. Whether the tagging of photos and videos or adding bits of text to extend the information and degree of knowledge about a certain content, annotating is very popular. But, as it has also been the case for the fragmentation and aggregation, arising from this multitude of places where annotations occur, many different styles of annotation next to methods of saving, managing, and modelling them exist. As a consequence, many annotations can only be used in the context they are created. Acting against this restriction, it is beneficiary to make use of a framework that handles this kind of problem by enabling rules and mechanisms for annotations in terms of its shape and creation.

In section 3.1.1, diverse standards have been mentioned which also had the idea of solving at least a part of this problem, but most of them covered only a small percentage of the available content.

[37] http://www.w3.org/TR/media-frags/
[38] http://www.w3.org/TR/2011/WD-media-frags-recipes-20111201/

Tackling these problems, the Media Annotation Working Group (MAWG[39]) developed the Ontology for Media Resources 1.0[40] [SBB+13]. The ontology currently incorporates 19 of the most commonly used and known metadata formats (for example Dublin Core, EXIF, MPEG-7, OGG, ...) and seven media container formats (Flash/FLV, MPEG-4/MP4, WebM, ...). By supporting a set of properties which specify the basic metadata that is needed to describe common multimedia properties, as well as a mapping that semantically connects existing vocabularies, divagating metadata standards are tied together. Their set of core properties is based on the current common multimedia schemas and span from 20 descriptive metadata properties (like identifiers, genre, ratings, language, contributors, or creation date) to eight technical metadata properties (like format, duration, or frame size). Descriptive semantics clarify the use of the defined properties. The mappings from their vocabulary are backed up by a subset of the SKOS vocabulary[41] to specify to what extent the property of the ontology matches the property of the given source model. The specifications are *exactMatch*, *broadMatch*, *narrowMatch*, and *relatedMatch*.

Next to the ontology, the Working Group also established an API[42] which is specified to global interfaces with certain parameters. Through this API, an interoperable access to metadata information on the web is enabled. It can be run in either asynchronous mode, where calls return without waiting for the request to finish their execution, or synchronous mode, which represents the counterpart.

The Open Annotation Data Model[43] (OADM) [HSSVdS11] is such a framework. It allows the creation of annotations that are easily shareable between platforms while trying to satisfy complex requirements and being as easy as possible at the same time. Starting from a simple base model - an annotation object that has a body and a target, both being connected, with the body resembling the annotation content and the target being the item that is to be annotated - adding different modules and information allows the annotation to be made more specific and fitting to certain use cases. There is no predefined protocol for transmitting the annotations, nor is there a defined way to store and maintain them. Transmission is established by supporting a web-centric method without the need for servers and clients, while the storage is covered by making available one serialisation that describes the annotation model itself. If there is the need for serialisations of other models, those rules should be supported as well.

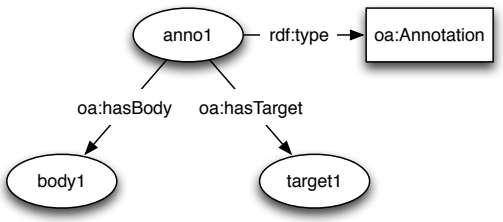
The OADM is made up of a core vocabulary that can be extended by different modules if needed. The modules are divided into *Specifiers and Specific Resources*[44], *Multiplicity Constructs*[45], and *Publishing*[46]. We refer to those specifications for further detail, as we will only cover the core functionality here to get a better understanding of how the model and its annotations are built up. The whole model makes use of an own namespace combined with vocabulary from different other commonly known, like Dublin Core, FOAF, RDF/RDFS, and SKOS.

As already mentioned above, the baseline of every annotation is the same. It circles around the three vocabulary items `oa:Annotation`, which specifies the base class, and the two relationships

[39] http://www.w3.org/2008/WebVideo/Annotations/
[40] http://www.w3.org/TR/mediaont-10/
[41] http://www.w3.org/2004/02/skos/
[42] http://www.w3.org/TR/2014/REC-mediaont-api-1.0-20140313/
[43] http://www.openannotation.org/spec/core/index.html
[44] http://www.openannotation.org/spec/core/specific.html
[45] http://www.openannotation.org/spec/core/multiplicity.html
[46] http://www.openannotation.org/spec/core/publishing.html

oa:hasBody and oa:hasTarget. This is depicted in figure 7. From there, you can enhance different

Figure 7 A basic annotation of the Open Annotation Model, adopted from the specification



structures of the annotation by adding specific classes and relationships. As side effects, annotations can be queried for the specific feature (for example you only want the annotations that have a special type in combination with a semantic tag), and this does also facilitate the process of interpreting the annotation at another time and place. For example, the body and the target can contain typing information by adding a class (for example dctypes:image, dctypes:sound) in conjunction with a rdf:type relation. Textual annotations can be treated a little differently. In other vocabularies, the text of the annotation is the content of the class itself. The OADM models it by using a node in combination with special classtypes (cnt:ContentAsText and dctypes:Text) and supports a format (dc:format) as well as the text content itself (cnt:chars) via relationships. Next to annotating content, tagging is also very commonly used. Tagging represents the process of adding only single words or short phrases to your resource. Semantic tagging, as a subtype of it, means the assignment of semantic concepts to your multimedia content, which can then lead to a semantic resource. This resource can contain even more information about it (for example, adding the semantic resource *http://dbpedia.org/resource/Spain* to a photo, additional information like the population, geolocation, and so on can be accessed automatically). In terms of shape, normal tags are similar to a text annotation, but are typed with oa:Tag, semantic tags are typed by the oa:SemanticTag combined with the foaf:page relation that points to the document which the annotation is to describe. By using the OADM it is also possible to annotate fractions of multimedia content instead of the whole item by supporting the fragmentation information in the URI of the target. This can be done similar to the fragmentation that we described in section 3.1.1. In specific cases it is useful to support annotations that have no body or annotations that have multiple bodies and/or targets, which is also provided by the model. This can be achieved by leaving out the body part of the annotation, or allocating multiple bodies and/or targets respectively. Another important part of the core specification is the provenance information that can directly be stored for every annotation. This is done by the relationships oa:annotatedBy, which describes an agent that does the annotation, and oa:serializedBy, when the annotation is generated automatically by software. Both processes are supported by a property that defines the point of time when the annotation has been created, which are oa:annotatedAt and oa:serializedAt. The agent can further be specified by being typed (rdf:type) as foaf:Person, foaf:SoftwareAgent, or foaf:Organization, and characterised more deeply by relationships or properties like foaf:name, foaf:openid, foaf:mbox, or foaf:homepage. The last part of the specification deals with the motivation of an annotation, the reason why the annotation has been created. This is established by connecting the annotation object with the relation oa:motivatedBy to an instance of the reason. These are for example oa:commenting, oa:highlighting, oa:replying, or oa:tagging.

OADM is a very generic model that can be used for various kinds of media types but is mainly built for manual annotation. The Stanbol Enhancement Structure (1) is used by Apache Stanbol (2) to represent results of an automatic content analysis process. This structure defines a set of different annotation types that all extend the basic type *Enhancement*. While the set of annotation types might be extended over time, this guarantees that every extracted features is described in a homogenous way. The role of the *Enhancement* concept is to provide provenance information by using properties of the DC terms ontology. In addition, it refers to the annotated content as well as other enhancements and provides the normalised confidence of the automatic process.

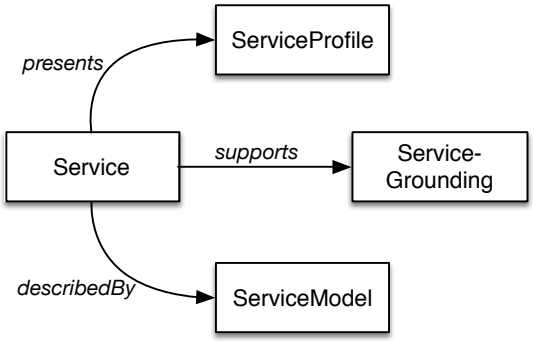
Stanbol is currently specialised on textual analysis and therefore provides a specific model for text annotations. The base type *TextAnnotation* is used to annotate sections by defining start/end character offsets as well as prefix, selected-text, suffix and the selection-context. It identifies textual fragments that can be linked to concepts like entity references. To express confidence values (how likely a concept is representing a string within the text), Stanbol uses the type *EntityAnnotation*. In the case of ambiguity, a *TextAnnotation* can be linked with multiple *EntityAnnotation*s to represent the different options. In MICO, we will mix up the Stanbol Enhancement structure with the ODAM to get a powerful model for contextual annotations.

3.2 Semantic Web-aware Description of APIs

Besides annotation of media resources, the processing units encapsulated in RESTFul services have to be described in a Semantic Web aware format. Those annotations create the basis for service orchestration handled by WP2. Following this, we will introduce three (standardised) approaches using Semantic Web technologies:

OWL-S. OWL-S [MBH+04] has superseded the DAML-S[47] specification and is intended to add a semantic markup to Web services. From a design point of view, it is created as an upper ontology based on the W3C OWL[48] specification. The main aim of OWL-S is to enable automatic discovery, invocation and composition of Web services, as well as the monitoring of their life cycles. Figure 8 depicts the composition of OWL-S, which will be discussed next: `Service` class is the most general

Figure 8 Composition of the OWL-S ontology



[47] DARPA agent markup language for services, `http://www.daml.org/services/`
[48] `http://www.w3.org/2004/OWL/`

description for a published instance of a Web service. This class holds the `presents` property meaning what the service does, `supports` property explaining how to interact with the service and `describedBy` property defining how to use it.

`ServiceProfile` class encapsulates information to automatically discover the features of a service. In addition to the feature description, the limitations of a service, quality of service parameter and requirements are also part of this document.

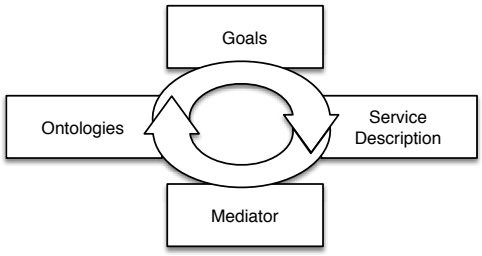
`ServiceGrounding` class gives details on how the service can be invoked. Here, a communication protocol, message formats and service-specific details (e.g., ports) are given. For each feature depict in the `ServiceProfile` class, the `ServiceGrounding` class defines the serialisation format of each feature as well.

`ServiceModel` class explains how the features of a service can be used by describing semantic content of requests, the conditions producing a result and the workflows.

OWL-S has been published as an official member submission in the W3C space.

Web Service Modelling Ontology. The Web Service Modelling Ontology (WSMO)[49] [RKL+05] is the result of joint research efforts of various EU projects. At its core, WSMO defines a conceptional meta-model and a formal language – *Web Service Modelling Language (WSML)* – to generate semantic descriptions of Web services. Besides those definitions, a reference implementation – *Web Service Execution Environment (WSMX)* – is available to perform dynamic matchmaking, selection, mediation and invocation of semantic web services based on WSMO. The WSMO ontology is split up in four

Figure 9 Composition of the WSMO ontology



main components depict in Figure 9: *Ontologies* formalise the actual domain knowledge and serve as the basis for inference. *Goals* model the users view in the offered Web service process and therefore describe a users expectation of a functionality. *Service Description* define technical aspects of the Web service. This includes its public interfaces, preconditions, in- and output parameter and its behaviour related to success or fault cases.

Mediators enable interoperability in terms of the data, process and protocol level. Here, mismatches can be resolved and mappings are specified.

[49] `http://www.wsmo.org/`

A major community concern in terms of the WSMO is that it has been designed isolated from standardisation bodies, such as W3C.

Hydra. Hydra [LG13] is one of the most recent attempts in the research community to enable the semantic annotation of Web APIs. Its main aim is to simplify the creation of RESTful APIs, where the incorporated resources are dereferenceable with IRIs.

In this regard, Hydra specifies a vocabulary to enable the server to advertise valid state transitions that can be consumed by client applications accordingly. On the basis of this information, the client is able to formulate new HTTP requests to get into another state or to trigger another operation. In the vocabulary, the `ApiDocumentation` class is most central and serves as the main entry to a service. Hydra's `Resource` class is a subclass of RDF `Resource` class guaranteeing its dereferenceability. For dynamically generated IRIs, the `IriTemplate` class specifies the characters and there composition for a legal IRI construction on the fly. At its base, it consists of `template` and `mapping` declarations. To invoke a service, the client retrieves `Operation` instances that define valid HTTP requests including `method`, `expects` and `returns` parameter. To simplify APIs, Hydra is pre-equipped with definitions such as for CRUD operations. A comprehensive overview of the vocabulary and usage examples can be found at the current draft of the specification[50].

Hydra is a recently started community project[51] in the W3C space and its members are currently actively working on an official specifications.

3.3 Trust

In order to maintain a functioning society in the real world, trust among the inhabitants of the world as well as trust within the several layers of society is of great importance [Fuk95]. It is a logical consequence that this holds also for online communities and market places. Further, trust is an essential part of the Semantic Web Stack and therefore a foundation for the Web of Data in terms of quality assessments [BHBL09]. There exist manifold definitions of the word trust. According to [AG07] the following definition of trust is suitable for its application to the MICO domain based on the interaction with data:

> "Trust of a party **A** to a party **B** for a service **X** is the measurable belief of **A** in that **B** behaves dependably for a specified period within a specified context (in relation to service **X**)."

The most common way to infer trust is realised by taking the information of identity and relationships into account. This approach is based simply on knowing someone [GPH03]. The resulting relationships can be visualised as a bi-directed graph, where a person is represented as a node and an edge embodies the relationship.
But not only in the aforementioned topics can trust enhance the results and systems altogether. Trust is very often found in combination with recommender systems, as people tend to rely more heavily on opinions and suggestions made by people or friends they trust, rather then depending on a recommendation solely made by a platform. This and the definition of trust is applicable for our purpose of recommending different items to a given user. Recommendation scenarios arising from the MICO project and use cases that can be enhanced by a trust-based component might be the following:

[50] `http://www.hydra-cg.com/spec/latest/core/`
[51] `http://www.w3.org/community/hydra/`

- The process of recommending an item **X** to a user **A** can be enhanced by increasing (or decreasing if distrust is also considered) the score of the item **X** depending on the fact, that a person or friend **B** of **A** has already liked (or disliked) the same item. In this case, the item **X** can be any content or advertisement item. The trust is represented by a trust-value between user **A** to **B**.

- Trust could also be applied in MICO's production chain, as trust and provenance will play a part concerning the extraction process and metadata publishing. Trusted extractors will deliver higher scores for the contemplated multimedia objects resulting in a higher relevance for the object to be found and consequently be requested by the multimedia querying step.

The decision for a trust model splits itself in different topics as the calculation of a trust value in one instance is segmented in multiple components. These will be covered in this section. Section 3.3.1 will depict the different possibilities of illustrating trust values, while section 3.3.2 extends the interpretation of trust by adding a "negative" component in the form of distrust. When a trust value cannot be assigned directly between the truster and its trustee (over one edge), third parties have to be included. Calculating trust over several edges requires the concept of trust propagation (section 3.3.4) to "forward" trust calculation and a subsequent trust aggregation (section 3.3.5) to sum up the multiple trust values into one single resulting value for the truster. For these two concepts trust computation and different metrics are applied, which are discussed in section 3.3.3.

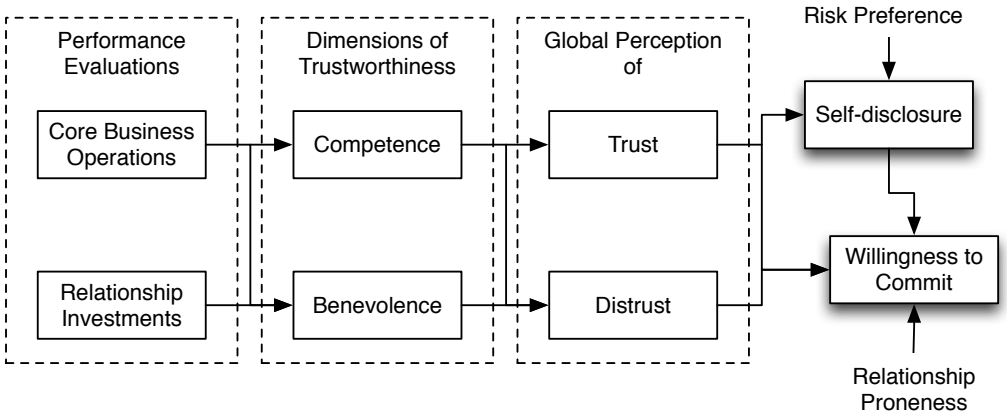
3.3.1 Trust Representation

In nowadays trust-based systems the representation of trust values between two users is expressed in one of two fashions: *probabilistic* or *gradual*. Probabilistic approaches assign trust values in a "all or nothing" manner and express, if a user **A** (the truster) can either trust another user **B** (the trustee) or not. This fact is determined by a probabilistic value that is calculated for the truster. Subsequently, the process computes a percentage to which extent the trustee **B** can be seen as trustworthy. [TPJL05] make use of such a probabilistic approach for their model by tracking and remembering successfully or unsuccessfully fulfilled interactions between a truster and its trustee. One interaction means that **A** wants **B** to deliver a certain service, the outcome then depicts if **B** fulfils its obligations or not, which corresponds to a value of "1" for a successful transaction or "0" otherwise. By weighing those two numbers against each other, the truster can calculate a probability for the trustee to commit to its deeds and as a consequence him or her being trustworthy or not. If the truster on the other hand does not have a history with the corresponding trustee, third parties have to be included in order to estimate the so called reputation of the trustee (this resembles the steps of trust propagation and aggregation, which will be covered in sections 3.3.4 and 3.3.5). With the help of that reputation value, the truster can infer a trust value for the trustee.
Gradual approaches to model trust differ from its probabilistic counterparts in the way that they do not "only" assign trust values in a black or white fashion, but rather depict the fact, that another user or a service can be trusted (or distrusted) to some extent. Humans paraphrase this circumstance by saying that someone else is for example "very trustworthy", "a little trustworthy", or "very untrustworthy" etc. [FPC03] implement this kind of model which is based on different beliefs done by the truster towards the trustee. These are picked from a range of internal and external attributions. Examples are ratings like competence, disposition, or harmlessness. By combining all beliefs that have been made for one attribution and assigning a weight to it (whether the attribution is considered important or not important in the given situation), a value of trustfulness can be calculated for the specific trustee at the given service or scenario. The belief values as well as the weights have a number range of [-1, 1] and are combined with Fuzzy Cognitive Maps (see [Kos86]) for the calculation.

3.3.2 Trust and Distrust

In many cases in the literature, trust models skip the concept of distrust, as it is mostly just interpreted as the counterpart of trust. According to this interpretation, if someone has high trust into someone else, the distrust is low or not existing at all. On the other hand, when a low trust value is present, distrust is considered high. But there is also evidence and research, that distrust can and even must be interpreted in another fashion. For example, [Cho06] states that trust and distrust differ both in factors that influence them as well as the asymmetric effects that they have on other properties. Their model can be seen in figure 10. Starting from the middle, their base concepts are *competence* and *benevolence*. The

Figure 10 A model of trust and distrust, adopted from [Cho06]



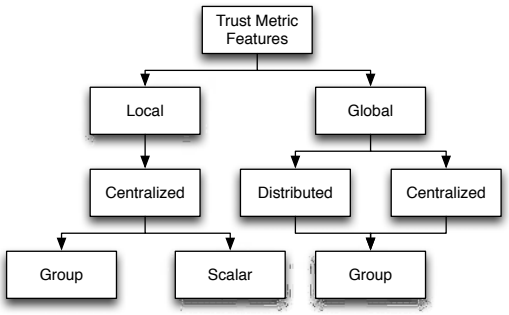
competence depicts the capability and reliability of a given person or service, while the benevolence is dealing with the aims of the trustee of being of good nature and not being interested in harming his or her partners. The influence of these two differ from person to person and they are also affected by ones judgement, goals, and attitude. For example, a negative value in benevolence has greater effects on the reduction of trust values while in comparison trust values are not increased in comparable scales as a result of a positive benevolence evaluation (this is also reassured by [SSS02]). Another important point are the factors, that trust and distrust themselves have impact on. In this case, the two main areas are *self-disclosure* and *willingness to commit*, which are closely related. Self-disclosure covers the trouble of giving some of your personal information free and making it accessible for others. As most transactions over the internet require at least a little bit of self-disclosure, trust is a very big concern here. Factors like the uncertainty about the behaviour of your trustee or trade partner as well as the vulnerability that you expose when committing to a transaction can constrain self-disclosure even further. An important thing to note here is, that the disparity of trust and distrust can be seen easily. If you assign a value to self-disclosure (whereas a high value stands for your willingness to commit to a trustee and reveal some of your personal information for the sake of a transaction), the value increases with a rising trust value, but not as much as it decreases by enhancing the distrust towards the trustee to an equal amount. The next step after committing to a single transaction is committing to a longer relationship with one trustee or a business partner. This is also influenced by self-disclosure, so consequently discrepancies between trust and distrust are present here, too.
Now that the clients side of trust and distrust is laid out, there are also features of the provider of

a service that will influence benevolence and competence criteria. Namely these are called the *core business operations* and the *relationship investments*. The former deals with things a venture has to maintain in order to stay active, known, and in business, for example product offerings, site designs, and security. Relationship investments cover ways with the effect of having customers that have done successful transactions coming back. Common strategies for this are the offering of rewards, frequent and personalised communications, as well as preferential treatment for good customers. Just like the other stages, these factors do have different leverage on others. As the relationship investments do create a higher benevolence for the customer, core business operations will convey a higher feeling of confidence.

3.3.3 Trust Computation and Metrics

The consideration and calculation of trust is very rarely a case of just two persons. People rely on opinions of their friends that already had interactions with your requested trustee, and if not, they might know someone who did. If that is not the case either, the third person him or herself might know someone and so on. In doing so, the calculation of a trust value towards one person becomes a transitive process over many other third parties. By taking this fact into account, a so called *web of trust* is built up. As already mentioned before, in this web of trust, the peers (whether these are people, services, providers of some sort, ...) are modelled as nodes while a trust relationship is an edge between two peers. Utilising this web of trust, customers can enhance their trust values about given trustees and they can overcome the problem that they might not have information about a specific peer they want to interact with (in this case, no edge would be present, so the truster does not have a trust estimation of his own yet). This is also called the *cold start problem*.
Inferring a trust value in such a web of trust is supported by two mechanics. Trust propagation (see section 3.3.4) shows techniques of propagating and distributing the trust calculation, while the aggregation step (see section 3.3.5) merges the results from the propagation step. Additionally to these two, trust metrics are needed, which define in what manner and what extent the web of trust will be utilised. There is an abundance of different metrics out there. Instead of covering all different specific metrics, [ZL05] give a very good classification of metrics by supporting different layers to distinguish them. They also provide a good variety of examples for their different metric classes. The classification can be seen in figure 11. The first layer divides the metrics into *global* and *local* metrics. Global trust metrics take into

Figure 11 A classification of trust metrics, adopted from [ZL05]



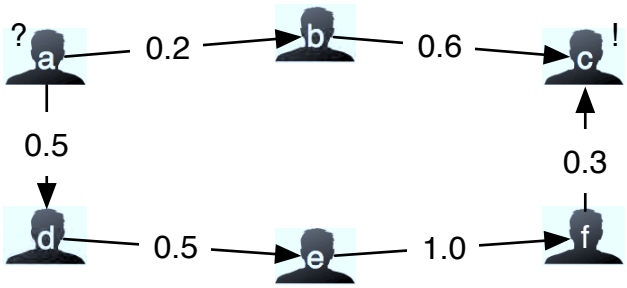
account every peer and trust relation that is present and reachable in the complete graph. Local trust metrics can be restricted to chosen smaller portions or subgraphs, which is based on personal bias. This

is encouraged by the fact, that third parties (over whom the trust value will be influenced) must not be considered trustworthy in the opinion of the truster itself. Those can then be excluded. An important point to consider is here, that if you intend to compute a more personalised trust value for a peer, local metrics should be chosen, because global approaches tend to result in a general reputation.
The second layer concentrates on the manner of computation. It is possible to calculate trust values in a *centralised* fashion. This means, that all trust information is gathered at the trusting peer and calculated in one big step. As a consequence, trust values and the information of all involved peers have to be accessible at all times, because it is necessary in order to do a full computation. Its counterpart is a *distributed* calculation, where the peers that are involved in the process can evaluate their own trust values (taking into consideration the same metrics and rules for propagation and aggregation that the base truster uses, as well as preceding trust calculations from other peers) and then only forward their own (partial) result. A distributed mechanic is only possible in global trust metrics.
Lastly, a distinction in terms of how links are evaluated is presented. *Scalar* metrics will compute trust values between every pair of two given peers **A** and **B** of a set **V**, so trust relations of subsets are also calculated. *Group* metrics on the other hand take more peers at once into the computation, which results in a trust value for a whole set of peers and consequently a more generalised trust value for that set. Considering all the points mentioned above, in a global metric background, only group evaluations are possible because of the overhead of a scalar trust calculation of every pair of peers of the whole amount of users (which in a common use case is considerably huge). Additionally, using a scalar trust metric gives more insight into the single trust values of the subsets of the calculation, which is accordingly the better choice for utilising a personally biased web of trust.

3.3.4 Trust Propagation

This section covers ways and modes that are possible to propagate the trust calculation through the chosen web of trust. But before this is possible, an important point to note here is the problem of the context. Trust relations are dependant on the context they base on, so if a trust-based recommender system focuses on more than one context, they should be treated differently. Consider the following example:
User **A** is looking for a good book and has a high trust value towards user **B**. User **B** himself does not know a good book to recommend, but he does have a trustworthy friend, user **C**, but their trust relation is based on film recommendations. Consequently, even if there is a chain of trusts from user **A** to user **C** over **B**, you cannot conclude that a recommendation for user **A** should be posed by the knowledge of user **C**, because of the different context. In the following we will assume an equal context base for our trust evaluation.
When a trust value is to be calculated for a trustee in a normal web of trust, generally you have more paths that lead from the truster to the trustee. An example is depicted in figure 12, showing six persons of a web of trust. User **A** is the truster, while user **C** illustrates the trustee. The trust values are also visible at the corresponding edges, which are directed from a trusting person to its corresponding trustee. The truster **A** wants to infer how trustworthy trustee **C** is. He or she did not have any past relationships with him or her, so the trust has to be calculated transitively over the web of trust. There is one path over user **B**, and a longer path over the users **D** through **F**. Both paths need to be computed separately by propagating the trust calculation through every path. After this is done, both values will be aggregated (see section 3.3.5). While there are different weights assigned to every trust edge (as users do have different levels of trust for each other), multiple trust propagation operators exist to deal with them. Those operators have different ranges of use and consequently need to be applied according to the specific use case. Among the easiest operators are the multiplication and fuzzy operators, for example

Figure 12 A simple propagation example with two paths from the truster to the trustee



the minimum operator, following the assumption that a trust chain is only as strong as its weakest link. Other possible propagation operators are based on fuzzy if-then rules (see [LB06], [WV03]), on the theory of spreading activation models (see [ZL05]), or the semantic distance between third party peers and a user's perception of their trust (see [AMCG04]). To clarify the impact of the proper choosing of a propagation operator, consider the multiplication and weakest link operators for the example seen in figure 12. While the multiplication provides an aggregated trust value of 0,12 for the upper and 0,075 for the lower path, the weakest link results in the values 0,2 and 0,3 respectively.
In addition to propagation operators there are several "modes" of trust propagation that infer different additional edges in the web of trust. In [GKRT04] (who we also refer to for deeper mathematical explanations), these are called *atomic propagations* and are namely:

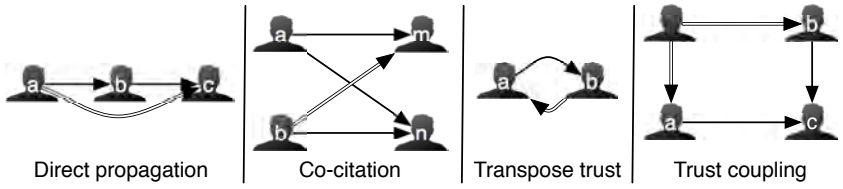
- Direct Propagation: This depicts the "normal" transitivity, so if user **A** trusts user **B**, and **B** trusts user **C**, it is inferred that user **A** should trust user **C**.
- Co-Citation: Stands for the conclusion of a forward-backward step in the web of trust. Imagine a user **A** trusting the users **M** and **N**, and a user **B** who trusts **N**. So this mechanic suggests that user **B** should also trust user **M**.
- Transpose Trust: The transpose trust mode can be compared to reflexivity, by inferring that if a user **A** trusts user **B**, then user **B** should trust **A** as well.
- Trust Coupling: If you trust a person **A** that shares the same trust relations with another person **B** (the common trustee user **C**), trust coupling suggests that you should trust user **B** as well, because **A** and **B** are very similar to one another.

Figure 13 illustrates all of the above mentioned atomic propagations. The examinations from above were done with the regard of only having to deal with trust values (and no distrust) as well as not considering the length of a trust path. Naturally you would think that the longer a path of trust is, its impact in terms of a final trust value should be diminished. In the literature this is called the *trust decay* (see [Guh03]). Additionally, the length of the paths considered in the computation can be restricted to a certain number or by algorithms like the shortest path. In terms of propagating distrust, we refer to [VDCC11] and [GKRT04].

3.3.5 Trust Aggregation

When multiple paths from the truster to the trustee exist, lastly they need to be aggregated into one trust value. The aggregation, similar to the propagation, supports simple operators like the minimum,

Figure 13 Illustration of the atomic propagations



maximum, weighted sum, average, or weighted average. These yield different results, so just like before, a fitting operator needs to be chosen according to the use case. Not much research is done in this direction, more complex operators are presented in [JMP06]. Influenced by a distributed trust metric, it is also possible to swap the order of propagation and aggregation, meaning that every peer can aggregate its incoming values first. They will propagate their partial result to the preceding peer instead of passing on all the trust values.

3.4 Provenance

Provenance is very well understood in the real world and the objects it contains. Provenance information gives the answers to questions like ”Who made this?”, ”Who edited it last?”, ”Where did it come from?”, or ”Who owns this?”. These questions are also applicable to ”objects” emerging from the world of data, and they can also hold and be responsible for viable information and decisions. In the MICO project, provenance information - similar to trust values that we discussed in section 3.3 - will play a part in the extractor chain. Possible results can lead to the early exclusion of extracted intermediary results due to a preceding step that is not considered as trustworthy or important, based on conclusions of its provenance information. This can lead to savings in terms of efficiency and lead to a much higher accuracy of the whole extraction, querying and recommendation process. Reliability, trustworthiness and quality of data is very often also related to provenance.

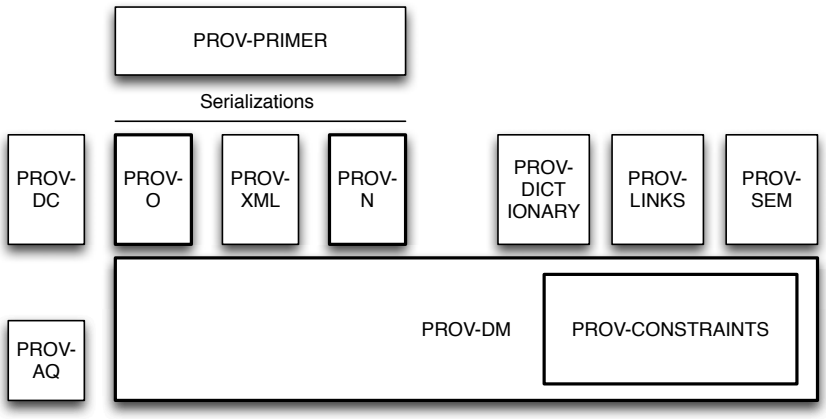
In chapter 3.1.2 we have already described in short the Open Annotation Data Model, which has a module that deals with provenance information. Furthermore, there are whole frameworks for the same purpose. One of them is the Open Provenance Model [MCF+11]. This framework is based on and refined by a series of challenges that have been posed on different workshops and conferences. The current version is 1.1. They designed their model in order to meet the following requirements, which have been adopted from [MCF+11]:

- To allow provenance information to be exchanged between systems, by means of a compatibility layer based on a shared provenance model.
- To allow developers to build and share tools that operate on such provenance model.
- To define provenance in a precise, technology-agnostic manner.
- To support a digital representation of provenance for any “thing”, whether produced by computer systems or not.
- To allow multiple levels of description to coexist.

- To define a core set of rules that identify the valid inferences that can be made on provenance representation.

Their provenance information and calculation circles around a provenance graph, which depicts a directed graph that expresses the dependencies and requirements for provenance conclusion. Three concepts form the base of the graph: artefacts, processes, and agents. Artefacts represent the objects, whether they are of physical or digital nature. The processes then symbolise actions or series of actions that are executed on or caused by artefacts, which then results in a new artefact. The latter concept illustrates an acting entity that is somehow related to processes (e.g. the enabling or controlling agent). From this point on, the objects are connected with different dependencies, that form the edges of the provenance graph. They are directed from the source to their target, and pose different inferences on the whole model that is depicted by one graph. There are also more specific dependencies or types of systems that can be covered using this framework, but for further detail, we refer to the specification itself. Another approach is posed by the W3C with PROV[52]. Their main focus is laid on the publication of provenance data as well as the access, validation, conversion and interchangeability between different vocabularies and information systems. Diverse formats like XML and RDF come to use, mappings exist for example to Dublin Core. The PROV is divided in ten documents or modules aiming at varying application areas. It is designed so that developers can start from a baseline and then adopt the PROV model according to their use case. All of the modules can work independently, so an overall knowledge of all of them is not necessary. Figure 14 shows the family of modules. For a deeper insight on every one of them, we advise to the specification[53] of the respective module. The PROV-PRIMER[54] contains the core

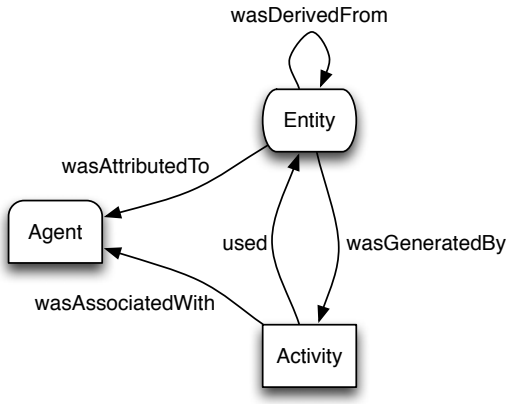
Figure 14 The family of modules for PROV, adopted from the specification



concepts that are needed for the PROV model, while PROV-O[55] defines a light-weight OWL2 ontology encoding of the PROV Data Model which is used for Linked Data and Semantic Web purposes. These will form the main interest for MICO. The PROV baseline model is constituted by several core concepts and elements. These are depicted in figure 15. In combination these concepts can form every requirement that can be posed by the provenance modelling that is underlying the data model. *Entities* are all

[52]http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/
[53]http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/
[54]http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/
[55]http://www.w3.org/TR/2013/REC-prov-o-20130430/

Figure 15 The base concepts of the PROV data model, adopted from the specification



kinds of ”things” that need to be modelled, for example data files, physical files, pictures, books, and so on. *Activities* depict the are the processes that can either create and generate or change the attributes of entities. They can also make use of already existing entities to create new ones. These activities and entities can now be attributed or associated to an *agent*. Agents are represented by real persons or they can also be software. In PROV you speak of an agent who is connected to an activity if you can address at least some kind of responsibility of that specific activity to the agent. He is then attributed to the corresponding entity that is the result of the process. Multiple agents can be associated with multiple activities or entities and vice versa. All the relationships taking part in the provenance modelling can also be enhanced by adding a *role*. This can be for example an agent that takes a specific role like a scientist annotating a picture (the annotation would be the activity while the picture symbolises the entity), or the role that is played by the entity, in our example the old picture that is used as base for the annotation. If an entity or its characteristics are at least partly related to another entity, PROV specifies it to be *derived from* another entity. Special forms of this derivation is the *revision* of documents and the *quotation*. When documents have different originating revisions, PROV creates a new entity for every revision that are related to one another. A quote is posed by a single link. Activities can be put together in a chain, which is named as a *plan*. *Timing* issues can also be modelled in PROV, which state at which point of time a document or file has been created or altered, for example. The last concept circles around the fact, that in a provenance modelling, some things can have different perspectives. In PROV, these are called *specialisations*. Specialisations are multiple entities that stand for one entity, having common fixed characteristica but differing in some special attributes. Examples from the specification contain a web page, that is altered over time. Every iteration of the altering process then creates an entity, that is a specialisation of the base web page.

4 Multimedia Querying

Responsible partner / Author: SRFG / Thomas Kurz
Related Technology Enablers: TE-401, TE-402, TE-403, TE-404, TE-405, TE-406, TE-407, TE-408, TE-409, TE-410, TE-411, TE-412, TE-413

One of the basic functions of a Database Management System is the efficient retrieval of stored data. The special needs of such a retrieval are strongly dependent on a) the stored data (and its underlying representation) and b) the specific use case. The MICO project focus on cross media data analysis, storage and retrieval and uses Semantic Web technologies for metadata representation (like described Section 3.1.2). Therefor the retrieval mechanism will be a mixture of classical multimedia functionalities and semantic web related data querying. Following the standard definition for Information Retrieval in [MR09] we define Multimedia Retrieval for Media in Context as follows:

> *Multimedia Retrieval on the Web of Data is finding (fragments of) resources of an unstructured nature (text, image, video, etc.) that satisfy an information need.*

whereby:

Web of Data means a dataspace of resources, which are represented in interchangeable, common formats, and interconnected by named links. Thus, the Web of Data is an exchange medium for data as well as documents, like described in the vision of the related W3C Data Activity group [W3C13]. The terms *Web of Data* and *Semantic Web* are used as synonyms in this document.

finding means providing a subset of web resources that meets someones expectations and is human-manageable in presentation form and amount (e.g. ordered list, collage etc.). This task includes the support of suitable ranking methods as well as pre-processing methods from data mining (e.g. clustering).

resources means in this context all things that are addressable via common web standards. For a seamless integration of Linked Data principles [BL06], information resources (metadata) must be accessible via HTTP protocol, non-information resources (video etc.) may use different (more suitable) protocols. In addition, the fragmentation of resources requires a suitable representation format, e.g. like the Media Fragments URI specification [TDMP12] described in section 3.1.1.

unstructured nature means that the resource is not interpretable per se but must be interpreted by experts or specialized machines to extract common understandable structure and features. This task is well supported for texts (e.g. Named Entity Recognition and Disambiguation [RT12a]) but is resource intensive for multimedia content. Due to the latest progress in cloud computing (e.g. map-reduce programming model) and the decreasing costs and dynamic accessibility of hardware, multimedia analysis is also supported for ”big content”. Some of these methods are described in Section 2.

information need means an abstract description of the expected subset or list. The more exact the information need is defined the more exact the presented set fits the expected results. The query language can be seen as an instrument for formalizing this need. It is an interface between user needs and the (mostly abstract) multimedia data and metadata storage layer. The more the language fits use case specific needs, the more adequate it is for the use case.

In this chapter we give a generic overview on both areas whereby we especially elaborate the different kinds of multimedia retrieval and point out the extendability of the most common query language in the Semantic Web called SPARQL. This gives us a good basis for the further progress in MICO workpackage 4 where we are going to extend SPARQL by capabilities for multimedia querying, including:

- spatio-temporal query operators,
- boolean as well fuzzy retrieval for annotated multimedia assets,
- multimedia specific operations like similarity search, and
- result aggregation and recombination for annotated media.

4.1 Multimedia Query Languages

The current landscape for Multimedia (MM) Query Languages is very broad and includes many different approaches. In this section, we categorize these approaches and highlight the features that enable a query language for multimedia retrieval. Additionally we give an introduction in querying the Web of Data because the combination of the two fields of classical Multimedia Representation / Retrieval and Semantic Web is a main goal we want to reach within the MICO project.

In the recent years there have been many query languages for multimedia retrieval. They can be classified in 6 main categories that are:

1. languages that extend SQL as the common standard for querying relational databases or follow an SQL-like approach, like WebSSQL [ZMWZ00] or SQL/MM [ME01],
2. languages that build or extend query languages for object oriented databases like MOQL [LOSO97] or POQL^{MM} [Hen01],
3. languages that are focusing an XML metadata structure, like MMDOC-QL [LCH01] or XQuery [BCF$^+$07] (which is not explicitly build for Multimedia),
4. visual query languages, like MQuery [DC96] (that focus on visual timeline retrieval) or Visual-MOQL [OOX$^+$99],
5. approaches that allow query-by-example, like [Jon07] or WS-QBE [SSH05], and
6. languages that try to build a meta-language, which are metadata agnostic and thus can be shared/distributed over several storage backends, like MPQF [DTG$^+$08].

Most of the above-mentioned multimedia query languages use proprietary metadata models to express descriptive information. Generally, this information is represented by XML instance documents based on a specific XML Schema (such as MPEG-7 [MKP02] or TV-Anytime [GS13]). For this purpose, one also needs to consider query languages that are designed for XML data queried by XQuery [BCF$^+$07]. The main drawback of XML is its limitations in expressing semantic meaning of the content information. This led to the development RDF, the basis of the Semantic Web as it is described in Section 3. Query languages related to RDF will be discussed in a later section 4.2. To get a clear picture of each category of multimedia query languages, we describe one example for every category in more detail.

4.1.1 SQL like approaches: MM/SQL

In the early 1990s the SQL (Structured Query Language) community came up with many incompatible extensions (especially for Multimedia) that forced the ISO subcommittee for SQL JTC1/SC32 to regularize such attempts. The proposed standard was immediately known as SQL/MM [ME01] and meant to integrate multimedia features to SQL. Like SQL, SQL/MM is a multipart standard that consists of various mostly independent parts. Part 1 [ISO00a] represents the backbone of the standard and describes, how other parts use SQL's structured, user-defined types required for the specific purpose of each part.

Besides multimedia functionality, text retrieval plays an important role for media in context. The full-text standard is covered by part 2 [ISO00b] and defines a number of structured user-defined types for storage. This is necessary because full-text in comparison to regular expression matching needs more complex data and query structures for (mostly language specific) tokenization, stemming, lemmatization, and fuzzy matching. In addition, fulltext search may support things like phonetic search (sounds like) and context search (heading, paragraph. etc.). Listing 1 shows a sample query using SQL/MM full-text extension on a sample table `documents` that includes a row `document` of type `FULLTEXT`.

Listing 1: Example for SQL/MM full-text search

```
SELECT * FROM documents
WHERE document.CONTAINS(
        '"dog" IN SAME PARAGRAPH AS
        SOUNDS LIKE "Balu"'
) = 1
```

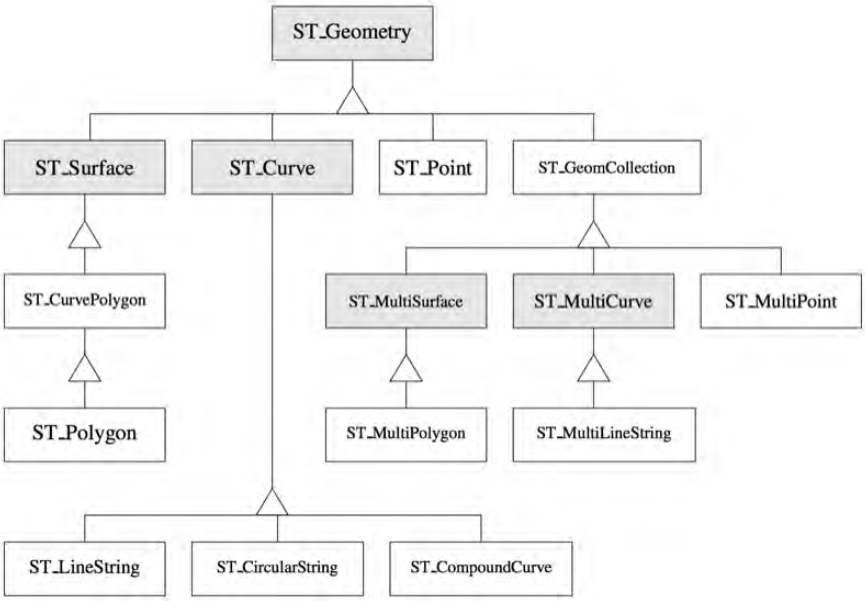
The query combines contextual with phonetic search to retrieve documents that most probably include a dog named "Balu", "Baloo", "Paloo", etc. This type of search can be useful in combination with automatic extraction techniques e.g. speech-to-text as described in Section 2.

Part 3 [ISO99] of SQL/MM covers the aspects of spatial data, such as geometry, location and topology. As described in [Sto03], SQL/MM defines a class model for 0- to 2-dimensional geometric objects (like points, lines, polygons or composites) as well as specific functions for spatial data. The spatial part of SQL/MM is mostly driven from geographic information system (GIS) but can be used for non-geographic use cases (e.g. fragment description for still images), too, whereby the reference system is replaced.

Figure 16 shows the SQL geometric type hierarchy for SQL/MM, which has been adapted from the geometric model of the OpenGIS Features Specification for SQL [Ope99]. The model differentiates between non-instantiable (supertypes) and instantiable types, like ST_Point, ST_Curve, etc.. There are many functions that can be performed over the spatial data model. They include the creation of new geometric objects out of existing ones, relational operations between objects like intersection or adjacence, and accessor methods that allow the extraction of fundamental information about type instance, e.g. the vertices of a line or the area of a polygone. Listing 2 shows a query that uses a spatial description of US counties to determine counties larger than the largest county in California[56].

[56] Sample is taken from `http://cs.ulb.ac.be/public/_media/teaching/infoh415/spatialnotes.pdf`

Figure 16 SQL/MM geometric type hierarchy [Sto03]



Listing 2: Example for SQL/MM spatial query

```
SELECT c1.county_name
FROM County c1
WHERE ST_Area(c1.geometry) > (
        SELECT max (ST_Area(c.geometry))
        FROM County c, State s
        WHERE s.state_code = c.state_code
        AND s.state_name = 'California'
)
```

The temporal aspects of multimedia were meant to be represented in part 4 of the SQL/MM standard but are not considered anymore, because *temporal* has a broader scope beyond the multimedia applications and thus is included in the revised SQL:2011 standard [ISO11], like described in [KM12].

In part 5 [ISO01] the standard focuses on storage, manipulation and retrieval of still images. The SI_StillImage datatype allows many formats (gif, png, tiff, etc.) for in- and output as well as for internal representation. The type also captures basic information about each image, such as format, dimension, color space, and so forth. Several operations can be applied on SI_StillImage including scaling, rotation, cropping, and shearing. SQL/MM also supports complex feature types, such as SI_ColorHistogram and SI_Texture (for coarseness, contrast, etc.).

In addition to classical multimedia features, SQL/MM also includes a part 6 about Data Mining [ISO06], but we consider it as out of scope for our project.

4.1.2 OQL like approaches: MOQL

Object oriented databases combine database capabilities with object-oriented programming capabilities. This type of database management systems has been very popular a few years ago. The effort has been mainly driven by the Object Data Management Group (ODMG) that came up with several specification components including an object model, an object definition language (ODL) and a declarative, nonprocedural language for object oriented querying and updating (OQL) [CBB+00]. With MOQL (M for Multimedia), this query language has been extended to deal with spatial, temporal and presentation properties by introducing new predicates and functions. In comparison to other approaches in the object oriented QL domain, MOQL is suitable for both video and still image retrieval. Most of the extensions of MOQL are placed in the WHERE clause in the form of 3 new expressions, namely *spatial_expression*, *temporal_expressions* and *contains_predicate*. Additionally, MOQL introduces a PRESENT statement that allows to specify how to deal with retrieval objects, especially with different mediatypes that has to be synchronized. We outline MOQL in this section because it has a clear focus and a user-friendly language design.

Contains predicate
The contains predicate is an relation between an instance of a particular medium type (e.g. an image) and a salient object which represents an physical object that is *contained* within the medium (e.g. a person). Listing 3 shows a query that aims to retrieve all images in which a person appears.

Listing 3: Example for MOQL contains query

```
SELECT m
FROM Images m, Persons p
WHERE m contains p
```

Spatial predicates and functions
Spatial predicates compare spatial properties of spatial objects (such as a region, a point, etc.) with each other. A predicate (e.g. inside) can only compare specific types of properties. For example can *nearest* only be applied to two points, whereby *cover* can only apply to a region and a point / line. Spatial functions compute attributes of an spatial object or a set of spatial objects. The query in Listing 4 shows both a spatial predicate *coveredBy* and spatial function *area*.

Listing 4: Example for MOQL spatial query

```
SELECT province, forest, area(forest.region)
FROM Forests forest, Provinces province
WHERE forest.region coveredBy province.region
```

Temporal primitives and functions
MOQL supports a set of 13 temporal relations that has been specified in [All83] and are widely accepted, which are *equal*, *before*, *after*, *meet*, *metBy*, *overlap*, *overlapedBy*, *during*, *include*, *start*, *startedBy*, *finish*, and *finishedBy*. In addition, MOQL supports several so called continuous media functions especially for video objects and their frame character e.g. *firstClip* or *next*. Listing 5 shows a query that returns the last clip in which a person appears from within a video v.

Listing 5: Example for MOQL temporal query

```
SELECT lastClip(
        SELECT c FROM v.clips c
        WHERE c contains p
        ORDER BY upperBound( c.timestap )
)
```

Presentation statement
MOQL allows to integrate all retrieved objects of different media types in a synchronized way by adding a PRESENT clause. These layout consists of a spatial layout, which specifies things like number of images etc., a temporal layout, which allows to specify things like temporal order and total length, and a scenario layout which allows also the usage of other presentation models or languages. Listing 6 shows a query that presents the result (an image of a car and a video showing the same car) in two different windows simultaneously.

Listing 6: Example for MOQL present query

```
SELECT m, v
FROM Images m, videos v
WHERE for all c in (
        SELECT r FROM Cars r WHERE m contains r)
        v contains r
PRESENT atWindow(m, (0, 0), (300, 400))
        AND  atWindow(v, (301, 401), (500, 700))
        AND play(v, 10, normal, 30*60) parStart display(m, 0, 20)
```

MOQL mainly focuses spatial and temporal relationships but lacks any kind of similarity or best-match queries. Other object oriented approaches have different focuses, e.g. POQL^{MM} [Hen01], which targets asset similarity based on low-level features (like SQL/MM part 5 [ISO01]).

4.1.3 XML-based query Schemas: MMDOC-QL

The emerging of MPEG-7 [MKP02] multimedia standard and its XML Schema datatypes in the late 1990s triggered attempts for XML-based media retrieval. For expressing audio and visual features, MPEG-7 defines so called Descriptors, for the relation and semantics between theses features the standard provides Description Schemes. Video scenes for example can be formalized by using `SegmentDecompositon` with type `SpatioTemporal`. As most XML query proposals had limitations regarding this type of documents, MMDOC-QL [LCH01] (Multimedia Document Query Language) was introduced, a language with multimedia constructs that is based on a logic formalism called path predict calculus. Queries in this calculus are equivalent to the identification of path predicates that are satisfied by the XML tree document. This formalism allows to describe also spatial, temporal and visual datatypes and relationships by utilizing MPEG-7s description of media fragments.

In MMDOC-QL there are 4 clause types:

GENERATE / INSERT / DELETE / UPDATE are building the operation clauses. They are used to

describe the logic conclusions in the form of allowed element and path predicates.

PATTERN clause describes the domain constraints of free logical variables (parts of the XML documents) by using regular expressions.

FROM clause defines the source (files).

CONTEXT clause is used to describe logic assertions about document elements in logic formulas (path predicate calculus). Within the calculus the language uses a logic form of XPath axis-operators with logical variables in the path formula (e.g. DIRECTLY CONTAINING).

Listing 7 shows and example query, whereby the path formula in the CONTEXT clause asserts that element "Segment" with id equal to %id contains element "SpatioTemporalLocator" (where the video objects are located during MediaTime %x). The form of %id is restricted by a pattern. The other lines in the CONTEXT part specifies the selection of %t; the GENERATE clause manages the output of the result as XML element.

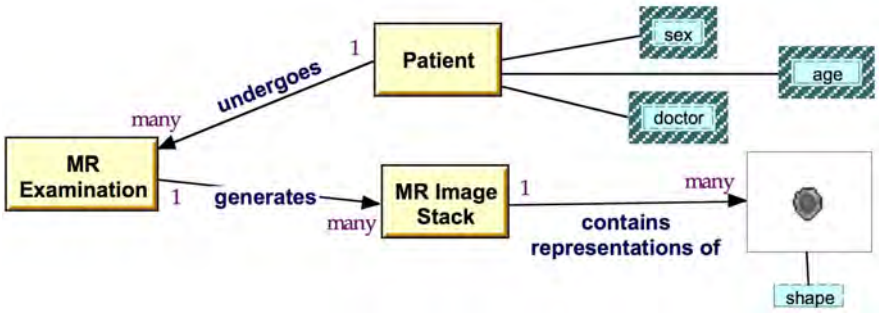
Listing 7: Example for MMDOC-QL query

```
GENERATE <List>
                <Videobject>%id</Videoobject>
                <ShowUpTime>%t</ShowUpTime>
        </List>
PATTERN {"MR"[0-9]/%id}
        {<region> ... </region>%focus}

FROM    mpef7video.xml

CONTEXT ( ( <Segment> WITH xsi:type="MovingRegionType"
        id=%id AT %movingregion )
        CONTAINING
        ( <SpatioTemporalLocator> DIRECTLY CONTAINING
                ( <MediaTime> AT %x ) )
        AND MEMBERP (%t %c)
        AND OVERLAP ( TRAJECTORY( %movingregion %t ) %focus )
)
```

4.1.4 Visual query languages: MQuery

MQuery is a visual query language for the domains of simulation and validation, medical timelines and multimedia visualization. The general framework that was worked out for querying all kind of multimedia data (images, sounds, long text, video, and timelines). The language has a direct, visual support for all these datatypes and includes the entire range of query operations (insert, retrieve, delete, update). It supports alphanumerical queries, multimedia results, multimedia predicates, time-based data, and also query nesting. Figure 17 shoes an example of MQuery for *obtaining the sex, age, and doctor of all patients with tumors similar in the shape to the tumor currently being viewed.*

Figure 17 MQuery: visual query example [DC96]



4.1.5 Query by example: WS-QBE

The visual database query language QBE (query-by-example) is a declarative query language. It is based on the relational domain calculus. WS-QBE is an extension of QBE and adds fuzzy logic concepts as well as a schema for query-weighting, which enables it for complex similarity queries in the multimedia domain. WS-QBE builds a core language for multimedia similarity queries but lacks specific features like spatio-temporal functions and predicates. Result presentation is not considered in the basic approach. Formulating a query in WS-QBE means to fill table skeletons. A query like "Find all oil paintings from a Dutch painter which are similar to a given image from my digital camera" is formulated by the two Tables 6 and 7.

Table 6 Query-by-example with WS-QBE: 1

painting	id	photo	painter	title	technique
P.		~ 	_painter		oil

Table 7 Query-by-example with WS-QBE: 2

artist	id	name	country
	_painter		Netherlands

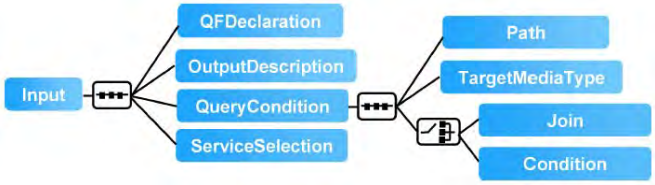
The table headings map the underlying database schema. By inserting one or more new tuples the user gives an example that is used for similarity calculation. The entry **P.** is used to indicate which fields (or tables) belong to the result set.

4.1.6 Generic Approaches: MPQF

The query languages we introduced are all strongly bound to the underlying metadata representation and schema. MPQF (MPEG Query Format) has the goal to unify the access to (distributed) multimedia repositories in a schema agnostic way. The Language specifies precise input and output parameters within XML documents but does not use specific elements that are related to a metadata schema like

MPEG-7 (like it is used for example in MMDOC-QL [LCH01]). An MPQF query always includes a *MpegQuery* root element with two child elements *Management* and *Query*. The management section provides a means for requesting service-level functionalities, the query section can either include an input or an output (depending if it is a request or a response). Figure 18 shows the schema diagram of

Figure 18 MPQF Input Query Format [DTG+08]



an MPQF Input element. It may contain one or more of the following elements:

QFDeclaration allows the definition of reusable definitions like paths and/or resources (descriptive as well as media resources) that can be referred from other parts of the query.

OutputDescription describes the structure and content representation for result set items. Furthermore it supports set operations like sorting, counting, and paging.

QueryCondition contains the actual filter criteria:

Path is a XPath expression and specifies the granularity of the retrieval, for instance if the process focuses on whole videos or on video fragments.

TargetMediaType contains MIME type descriptions like *audio/mp3* (if the user wants to retrieve audio files in MP3 format).

Join / Condition supports further diversity in filter criteria with arithmetic / boolean expressions, several query types (query-by-media, query-by-freetext, etc.) and joins.

ServiceSelection specifies a set of multimedia query services where the query should be evaluated.

Figure 19 MPQF Output Query Format [DTG+08]

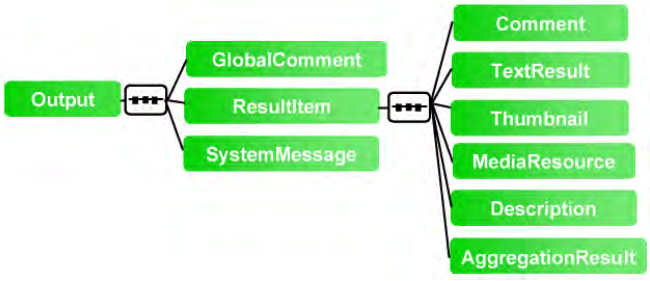


Figure 19 shows the schema of an MPQF Output element, which may contain one or more of the elements:

GlobalComment is meant for sending general messages such as the service subscription expiration or messages that are valid for the whole result set.

ResultItem element holds a single record of a query result with attributes *recordNumber*, *rank*, *confidence* and *originID* and the elements:

Comment is similar to GlobalComment but focus in the specific result item.

TextResult element holds the result item as type text.

Thumbnail carries the URL of a thumbnail image.

MediaResource carries the URL of the media resource in the requested format.

Description is a container for any kind of metadata in any format like MPEG-7 or TV-Anytime.

AggregationResult allows schema-valid result aggregation operation (e.g. SUM).

SystemMessages includes special messages regarding the responding system such as warnings or exceptions.

Listing 8 shows an example of a simple MPQF query that combines free-text search and conditions over XML metadata. The aim of the query is to find large images of "Hong Kong" (greater than 1000 pixels in width).

Listing 8: Example for MPQF query

```
<MpegQuery>
  <Query>
    <Input>
      <OutputDescription thumbnailUse="true">
        <ReqField typeName="MediaInformationType">
           MediaProfile/MediaFormat/FileSize</ReqField>
        <ReqField typeName="CreationInformationType">Creation</ReqField>
      </OutputDescription>
      <QueryCondition>
        <TargetMediaType>image/*</TargetMediaType>
        <Condition xsi:type="AND" preferenceValue="0.1">
          <Condition>
            <FreeText>Hong Kong</FreeText>
          </Condition>
          <Condition xsi:type="GreaterThanEqual">
            <ArithmeticField typeName="MediaInformationType">
              MediaProfile/MediaFormat/Frame@width
            </ArithmeticField>
            <LongValue>1000</LongValue>
          </Condition>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>
```

Further examples for MPQF queries can be found in [DTG+08].

The query languages described above support various MM specific features. Some of them try to cover all of them (e.g. [DTG+08]) and some are specialized on a specific one (e.g. [SSH05]). Based on that the main features we are going to focus for MM retrieval in the MICO project:

Query-by-keyword specifies a pattern query on freetext fields. This query uses similarity metrics like Levenshtein distance to compare string similarity. Both, the query- and the fieldtext can be preprocessed, e.g. stemming, lemmatisation, stopword elimination etc..

Query-by-example specifies a similarity or exact-match retrieval, whereby the query itself is a multimedia content (image, sketch, video, text etc.). The distance measure can include low-level (e.g. color histogram) as well as high-level features (e.g. semantic relatedness of describing features).

Query-by-spatial-relationship includes spatial relation like neighborhood (e.g. if object A is left beside object B) and/or spatial aggregation like bounding box within the query. In case of text, this can also be word spacing.

Query-by-temporal-relationship includes temporal relation like neighborhood (e.g. is object A appears after object B) and/or temporal aggregation like intermediate space. It is self-evident that the media content has to have a temporal component.

Query-by-relevance-feedback specifies a iterative retrieval process that take into account the results of a previous search, which are rated as good or bad by users. This type is strongly related to the query-by-example.

Query-by-media-function includes operations on media assets that add a higher level of semantics (e.g. a face recognition function) or functions that build fragments for further use (e.g. extract audio from a video item).

4.2 Semantic Web Query Languages

Depending on the underlying data format there are three main categories for Web Query Languages, as described in [BFS05], namely XML Query and Transformation Languages, RDF Query Languages and Topic Maps Query Languages. In the case of Semantic Web as described above only the RDF ones are relevant. RDF Query Languages can be grouped mainly into seven families that differ in aspects like data model, expressivity, support for schema information, and kind of queries. The families are RQL [KAC+02], XPath-, XSLT-, and XQuery-based Languages (e.g. [Sch04]), Metalog [Mar04a], Reactive Languages like [Pru04], Deductive Languages like [DSB+05], and, in the sphere of Linked Data, Path Traversal languages (like SQUIN [Har13] or LDPath [SBK+12]), and the SPARQL family with its most common instance SPARQL (SPARQL Query Language for RDF) [HS13a].

The SPARQL query language for RDF (SPARQL) SPARQL is an extension of RDQL [Sea04] and provides Semantic Web developers with a powerful tool to extract information from large datasets. It is designed to meet the use cases and requirements identified by the RDF Data Access Working Group. SPARQL allows expressing queries across diverse data sources, whether the data is represented as RDF. A formal description of SPARQL and its semantics by transform SPARQL into the relational algebra is described in [Cyg05] and [PAG09]. The query language is a syntactically-SQL-like language for querying RDF graphs via pattern matching. It includes features like basic conjunctive patterns, value filters, optional patterns, and pattern disjunction. In addition to the query Language itself, the W3C

recommendation also specifies a transfer protocol, a description for SPARQL Services, and several query result formats. In the next sections we describe SPARQL, whereby we especially highlight the extendability we want to utilize within the MICO project.

4.2.1 SPARQL Protocol and RDF Query Language

SPARQL defines a standardization for RDF query syntax, semantics and protocol. It allows interoperability on the level of expressing rich queries on RDF datasets. The SPARQL Standard 1.1 Recommendation is separated in 11 parts [57], whereby the most important ones are the data retrieval language SPARQL 1.1 Query Language [HS13a], the data manipulation language SPARQL 1.1 Update [GPP13], the definition of the results formats with their most important representative SPARQL Query Results XML Format (Second Edition) [Haw13], and SPARQL Protocol 1.1 [FWCT13], a means for conveying SPARQL queries and updates to a SPARQL processing service and returning the results via HTTP. In this section we introduce the SPARQL 1.1 query language by highlighting some details.

SPARQL follows an SQL-like syntax but is based around graph pattern matching. Smaller patterns can be combined to complex graph patterns in various ways. The 4 main types of queries are SELECT (which returns a result table), CONSTRUCT / DESCRIBE (which returns RDF triples) and ASK (which returns a boolean value). Basically, a SPARQL query may consist of one or more of these clauses:

PREFIX allows to shorten URLs.

SELECT / CONSTRUCT / DESCRIBE / ASK is the projection clause. It identifies the return values, mostly variables that are bound within the where clause. Additionally, aggregation functions like AVG, SUM, etc. or custom ones are often used here.

FROM / FROM NAMED identifies the subgraph that is used to calculate the results. This enables SPARQL not just for querying triples but also quadruples.

WHERE is the selection clause. It identifies the values and bind the variables for the projection. Several constructs are allowed within the where clause, e.g. OPTIONAL, UNION, FILTER, negation, etc.

LIMIT / OFFSET / ORDER BY are sequence modifiers that can be used to change the quantity and the (per default random) order of a result set.

GROUP BY / HAVING are used to aggregate results, whereby HAVING is similar to FLTER in a WHERE clause.

For the matter of readability the list of clauses is not complete but includes the widely used ones. Like in SQL, in SPARQL 1.1 subqueries are allowed, too.

Listing 9 shows a simple SPARQL query that selects first- and lastname of persons having a lastname that starts with 'A', ascendent ordered by their age.

[57] http://www.w3.org/TR/sparql11-overview/

Listing 9: A simple SPARQL query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX sample: <http://example.org/sample/>

SELECT ?firstname ?lastname
WHERE {
        ?p a foaf:Person.
        ?p foaf:firstname ?firstname.
        ?p foaf:lastname ?lastname.
        ?p sample:age ?age.

        FILTER regex( ?lastname, "^A" )
}
ORDER BY ASC( ?age )
```

Variables in SPARQL are marked by the use of either ”?” or ”$” followed by a string of characters; the ”?” or ”$” is not part of the variable name. Variables are bound within the WHERE clause, the most important pattern, which is a kind of group graph pattern. SPARQL 1.1 defines some functions for filtering and aggregation (e.g. regex), which can be extended with custom operations.

Since version 1.1 SPARQL also takes into account the trends towards Linked Data and supports path expressions within patterns (whereby a triple pattern is also a special path expression of length 1). Listing 10 shows an example of a property path including an alternative path with an arbitrary length match. Such a fact is not expressible with simple triple patterns.

Listing 10: A SPARQL path expression

```
{ ?ancestor (ex:motherOf|ex:fatherOf)+ <#me> }
```

Path expressions also support some forms of limited inferences, for example for RDFS, all types and subtypes of a resource, like outlined in Listing 11.

Listing 11: Simple inference with SPARQL path expression

```
{  <http://example/thing> rdf:type/rdfs:subClassOf* ?type }
```

4.2.2 SPARQL Extensions

Responsible partner / Author: UP / Kai Schlegel

Although SPARQL 1.1 provides a powerful feature set, it also offers different possibilities for custom extensions to support specific purpose functions. There can be many use cases where SPARQL should be able to exceed the standard function set. Examples among many include full-text search, temporal and geospatial distance functions, multimedia similarity or custom aggregation functions. To embed custom function into the fundamental idea of declarative programming of SPARQL four main possibilities to extend SPARQL exist:

Filter Functions Extension
SPARQL query language allows filtering of query results through predicate expressions. Custom filter functions allow the definition of additional operations in the FILTER operator of SPARQL. The functions are globally identified by an IRI and take some number of RDF terms as arguments. Listing 12 shows an example how filter functions can be exploited to allow geospatial distance functions in SPARQL.

Listing 12: SPARQL Filter Function example

```
FILTER (custom:geoDistance(?placeA,?placeB) < 10)
```

Filter functions are described explicitly by the SPARQL 1.1. specification under the name of *"Extension Functions"* [HS13b] . Most of the popular SPARQL engines support extension functions, but the specification does not state how filter function should be implemented by the engine. The engine has to have a function registry to identify the correct implementation. In Virtuoso, for example, a developer can specify a extension function using SQL stored procedures or call nativ C code[58]. As a consequence the implementation of custom functions are globally not consistent and depend highly on the underlying SPARQL engine. An approach by Gregory Todd Williams suggests the possibility to implement extension function using a scripting language like JavaScript [Wil07]. The function itself is identified by a dereferencable HTTP URI. The SPARQL endpoint can resolve the URI and retrieve and execute the responded JavaScript code. In this way, the implementation of a function can be shared and retrieved on-the-fly.

Function Predicates Extension
Function Predicates, also often called Magic Predicates, Computed Properties or Property Functions, are basically similar to the mentioned filter functions. A predicate, identified by an IRI, denotes a custom behaviour. But instead of using the function in a FILTER operator, function predicates use the known convention of triple graph patterns. A function predicate is a predicate in a SPARQL triple that produces bindings using a stored functions instead of subgraph matching of existing triples in the database or inferencing. Many existing SPARQL engines support common functionality like free-text search exposed as function predicates. As an example, listing 13 shows how ARQ SPARQL processor could be queried for subjects which have a label starting with the string "Michael"[59] using the function predicate *textMatch*.

Listing 13: SPARQL Function Predicates example

```
PREFIX fp: <http://jena.hpl.hp.com/ARQ/property#>
SELECT * {
        ?subject fp:textMatch 'Michael*'
}
```

This overloading of predicates isn't explicitly mentioned in the SPARQL 1.1 specification, but implicitly allowed by sticking to the SPARQL grammar. In contrast to filter functions, predicate functions only allow 2-ary functions to be fully compliant with the SPARQL triple specification. Strictly

[58] http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtTipsAndTricksGuideCustomSPARQLExtensionFunction
[59] http://jena.apache.org/documentation/larq/

speaking, function predicates with multiple parameters are rather SPARQL language extensions. Some function predicates overcome this syntactic problem by using a RDF list in the object portion of the triple, because RDF lists don't have to be homogeneous (the members do not have to be of the same type).

Meta Extension
In a way, meta extensions depend on the aforementioned extension types. It allows the definition of new SPARQL functions and function predicates based on other SPARQL expressions or reusable SPARQL query templates. As a consequence, the SPARQL syntax and grammar does not need to be modified. Only the underlying SPARQL engine has to be aware of the processing of the SPARQL translation. SPARQL Inferencing Notation (SPIN)[60], a W3C member submission[61], forms the de-facto standard for SPARQL meta extensions. It offers a very powerful meta-modeling mechanism and allows user-defined functions which are based on SPARQL rules or constraints. For example, SPIN constraint can be used to link a RDF class with SPARQL queries that formalize invariants for the members of that class. In more detail, the RDF class "Square" could be automatically linked to all instances which have equal values for the "width"- and "height"-predicate. Meta extensions build a lightweight possibility to define new function, stored procedures or constraints based on existing SPARQL features.

Language Syntax Extension
The most powerful extensions of SPARQL can be done with real syntactic extensions. This kind of extension includes the modification of the SPARQL grammar and semantics by introduction new keywords or operations. For example f-SPARQL, proposed by Cheng, Ma and Yan [CMY10], introduces the fuzzy set theory to efficiently compute top-k answers based on user-defined weights. Another approach by Siberski, Pan and Thaden [SPT06] describe a SPARQL extension which directly supports the expression of preferences to allow scoring and ranking for result sets. Complex spatial and temporal SPARQL queries are supported by the SPARQL-ST extension [PJS11]. An example of the SPARQL-ST syntax is highlighted in listing 14. This query uses a SPATIALFILTER involving the inside function to ensure that geographical point falls within the given geographical area.

Listing 14: SPARQL Syntax Extensions example

```
SELECT * WHERE {
        ?c stt:located_at %g.
        SPATIAL FILTER (inside(%g, GEOM(POLYGON ((
                -75.14 40.88, -70.77 40.88, -70.77 42.35,
                -70.77 42.35, -75.14 42.35,
                -75.14 42.35, -75.14 40.88))))
        )
}
```

Obviously, the modication of the SPARQL syntax features a extensive way to add custom behaviour to SPARQL but does not ensure the interoperability and portability between different SPARQL endpoints.

As mentioned, efforts for standardization and interoperability are a crucial task for SPARQL extensions. The **SPARQL Extension Description**[62] by Leigh Dodds addresses this issue with a first

[60] http://spinrdf.org/
[61] http://www.w3.org/Submission/2011/SUBM-spin-overview-20110222/
[62] http://www.ldodds.com/schemas/sparql-extension-description/

draft of a small vocabulary for describing SPARQL extensions. Every extension function which is associated with an URI should be dereferencable and respond with metadata about the function itself. This information is useful to support interoperability, validation, documentation and automatic feature detection for SPARQL clients. The recently issued W3C recommendation **SPARQL 1.1 Service Description** [Wil13] provides a vocabulary for describing SPARQL endpoints in terms of supported features, available dataset and extension functions. The service description should be available by dereferencing the SPARQL endpoint URI using the HTTP GET operation. The vocabulary include the indication of extension functions or property features which may be used in a SPARQL clause like SELECT and FILTER. However, both approaches almost only describe which features are supported, but does not give any detailed information about the feature which is identified by an URI. Detailed information such as input parameter description or mapping between existing different extensions could be useful to improve the interoperability and mobility of SPARQL extensions.

5 Multimedia Recommendations

Responsible partner / Authors: UOX / Grant Miller, Chris Lintott; Zaizi / Rafa Haro; IO10 / David Riccitelli, Andrea Volpini

5.1 Introduction

In work package 5, we will investigate how the outcomes of cross-media extraction, cross-media metadata publishing, and cross-media querying can be used for computing recommendations for media content. The recommendation system will use standard approaches for computing recommendations and clustering (for example, feature vector similarity or k-means clustering). The main results will be:

- a standard model for transforming the multimedia metadata and analysis results into features suitable for recommendation and clustering (e.g. based on querying)
- a prototypical implementation of different algorithms for recommending and clustering media objects as a module for Apache Marmotta

The implementation of cross-media recommendations will be used as the main user-facing functionality in the use cases of the project.

Recommender systems (also called recommender engines or recommender platforms) have become very popular in recent years. Prominent examples include e-business sites with product recommendations like Amazon and Netflix[63], and social media sites like last.fm, Pandora, Facebook, and LinkedIn with related content and related user recommendations. Also, recommender systems are increasingly important in the news and media sector: for example, recommendations in Google News make up 38% of the click-through rate.

In general, there are three kinds of approaches to recommendations:

- Collaborative filtering is based on collecting and analysing large amounts of information about user behaviour, activities, or preferences and predict what users will like based on their similarity to other users; this approach is e.g. used by the Amazon product recommendation.
- Content-based filtering is based on the content and metadata of the items that are recommended and recommends other items with similar features or traits; features can include both features extracted directly from the content (e.g. instruments in a piece of music), and features explicitly represented in the metadata (e.g. likes or ratings of users); this approach is e.g. used by last.fm.
- Hybrid approaches are based on a mixture of different methods, including collaborative and content-based filtering; in many cases, these approaches have proven to perform better than the pure approaches; hybrid approaches are e.g. used by Facebook and Netflix.

[63] While these algorithms are usually proprietary, Neflix used a crowdsourcing competition to identify the most accurate recommendation algorithms. Details of the winning solutions are given here: `http://www.netflixprize.com/community/viewtopic.php?id=1537`. However, changes to the business (primarily the increase in streaming as a delivery model) and the need to encourage diversity as well as accuracy (in order to account for shared accounts across households) meant that this algorithm was not implemented (`http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html`). However, it remains a useful benchmark.

All kinds of recommendation systems have their specific advantages and disadvantages. For example, collaborative filtering typically suffers from a cold-start problem, sparsity of data, and scalability issues. On the other hand, content-based filtering depends heavily on proper understanding of the meaning of the analysed content and suffers from the so-called "Semantic Gap" problem. A semantic gap is the difference in meaning between two representations of the same concept, for example across languages or between a human and a computational system's use of the same term. Beyond these general approaches there are also more focused methods that e.g. take into account demographics or context. These can be considered as specialisations of the aforementioned approaches. Recommendation algorithms for all three approaches are well known and widespread. For example, typically used algorithms include cluster analysis (k-means, k-next-neighbours) or classification (bayesian, neural networks, support vector machines).

The Open Source framework "Apache Mahout" implements a whole collection of algorithms for free use. The main challenges are the quality of extracted features and scalability. Cross-media or cross-domain recommendations have also been investigated recently. In principle, cross-media recommendation does not differ much from the single-domain approaches. The main challenge here is in finding common traits across different media types and platforms. A further complication is in situations (such as that discussed for citizen science below) where the objective lies not only in providing the best possible recommendation for a user or customer, but in distributing recommendations evenly. In the case of citizen science, this means ensuring an entire data set is classified, but it might also apply to businesses adopting a sales model similar to Amazon but who are unwilling or unable to carry stock for a long period, meaning that it becomes desirable to exhaust the inventory.

5.2 State-of-the-art for generic recommender systems

Related Technology Enablers: TE-501, TE-503, TE-506

5.2.1 Introduction

In recent years, the interest in recommender systems has dramatically increased. Besides the typical e-commerce example [SKR99], where recommender systems are applied to increase the revenue by suggesting to the users items that they are might interested to buy, this research field has gained attention in many other domains where users usually suffer of information overload problems or where simply there is an interest on maximizing the fidelity of the users by increasing the time they spend on a site. Probably, Amazon.com is the most famous story of a recommender system successfully applied for e-commerce ([LSY03]). Based on purchase history, browsing history, and the item a user is currently viewing, they recommend items for the user to consider purchasing. As for Amazon, for many other highly rated Internet sites like NetFlix, Youtube, Linkedin, Yahoo, TripAdvisor, Expedia and so on, recommender systems are a key component.

5.2.2 Collaborative Filtering Systems

Recommender systems emerged as an independent research area in the mid-1990s when researchers started to focus the recommendation problem as modeling ratings estimations for the items that have not been explored before by the user. In these systems, user behavior is intended to be modeled as items' ratings. The most straightforward way to obtain the ratings is by explicitly letting the users to give a score to the items. Then, recommendations algorithms will try to cluster users with similar tastes

and will try to find relationships between items based on those ratings. In this way, recommendations depend always not only on the behaviour of a concrete user in the system, but also on the behaviour of the rest of the users. This approach has been widely known as Collaborative Filtering [SFHS07]. Collaborative filtering is considered to be the most popular and widely implemented technique in recommender systems. The key idea is that the rating of a user A for a new item I is likely to be similar to that of another user B, if A and B have rated other items in a similar way.

Several Collaborative Filtering algorithms have been designed and successfully implemented in the last years. User-based collaborative filtering, also known as KNN collaborative filtering, was the first of the automated Collaborative Filtering methods. It was first introduced in the GroupLens Usenet article recommender [RIS+94]. These systems evaluate the interest of an user for an item using the ratings for this item by other users that have similar rating patterns. This relies on a clustering process of users with similar tastes based on common and correlated ratings. As any other clustering algorithm, the key feature is the similarity function. For user-based approaches, a function for computing the similarity between users based on their preferences and profiles is needed. In [ERK11] it is possible to find a good survey of widely adopted similarity functions like Pearson Correlation, Constrained Pearson correlation, Spearman Rank Correlation or Cosine.

Item-based collaborative filtering [SKKR01] predicts the rating of an user for an item based on user's past ratings for similar items, being then two items similar if several users have rated these items in a similar way. Due to scalability problems of the user-based approaches when the number of users grows, item-based systems are nowadays more extended and adopted. Rather than using similarities between users' rating behavior to predict preferences, it uses similarities between the rating patterns of items. If two items tend to have the same users like and dislike them, then they are similar and users are expected to have similar preferences for similar items. As for users, similarity functions like Cosine and Pearson Correlation have been probed to be effective for computing items similarities.

5.2.3 Content-based Systems

Content-based recommender systems [PS13] learn to recommend items that are similar to those that the user liked in the past. The similarity of items is calculated based on the features associated with the compared items. The basic process performed by a content-based recommender consists in matching up the attributes of a user profile where preferences are stored, with the attributes of new interesting items. For example, if an user has listened before a set of rock songs, the system could suggest to listen new rock or similar styles songs. The most straightforward implication of this kind of systems is that they are completely domain-dependent. While for adopting a collaborative filtering system only a rating or preference model is needed, with content-based systems it is necessary for each domain to properly profile items and customize the similarity functions. For some domains, especially those involving unstructured items like multimedia files, item profiling is not an easy task resulting in weak item's representations. In these cases, content analysis techniques are applied in order to extract items' features and make an structured representation of them. For instance, when it is desired to recommend textual information, classic information retrieval techniques can be applied to extract features like term frequency, part of speech tags or n-grams, but also more advanced analysis process can be used to identify relevant concepts, categorize the content or extract underlying knowledge.

In the context of MICO, the outcomes of cross-media extraction, cross-media metadata publishing and cross-media querying can perfectly be used for improving item profiling in domains where

media content need to be delivered. Cross-media extraction software packages could be used to automatically extract media files features by analyzing the content. These features would be stored as metadata along with the manual metadata accompanying the content. Finally, cross-media querying can be used to match metadata of content liked by the users in the past with metadata of new interesting content. This recommendation process would be possible because user's preferences, content and metadata would coexist in the same knowledge graph, queryable using standard query languages like SPARQL.

After profiling the items, users are also profiled by collecting the items liked and not liked in the past. Users' profiles are typically used then to infer a model of users' interests able to give a score for each pair of user and new item. Recent content-based recommender systems use machine learning to rank techniques to train such models using the harvested preference data [MKP03]. Apart from the limited content analysis problem, another well known shortcoming of content-based systems is the overspecialization. Content-based recommenders have no inherent method for finding something unexpected. The system suggests items whose scores are high when matched against the user profile, hence the user is going to be recommended items similar to those already rated. Because of these shortcomings, it is rare to find a pure content-based implementation. It is more common to use hybrid approaches involving collaborative and content-based. The current trend in content-based filtering is to add social information to the items attributes, such as tags, comments, opinion, and social network sharing. Social tagging systems are the most popular because they allow users to annotate online resources with arbitrary labels, which produces rich information space filtering techniques.

5.2.4 Context-aware Systems

The majority of existing approaches to recommender systems focus on recommending the most relevant items to individual users and do not take into consideration any contextual information, such as time, place or surrounding events and trends. In other words, traditionally recommender systems only take into account two types of entities, users and items, and do not put them into a context when providing recommendations. However, in many domains, contextual information is important in order to deliver more accurate suggestions. For example, temporal information is an important factor for traveling as location is for restaurants recommendations. Also, for music recommendations in applications like Spotify or Pandora, it is important to take into account the current mood of the user.

[RRS11] provides a good survey about context-aware recommender systems, including techniques for modeling contextual information and including it in the recommendations algorithms. The most common way to use the contextual information is by introducing it as a new element in the recommendation process, just as users and items. In this way, the users' preferences don't longer consist on items they liked or rated in the past, but they now consist on a rating within a concrete context. For example, movies' preferences would also depend on where, what time and with whom the movie has been seen. As we can see from this example and other cases, the contextual information can be of different types, each type defining a certain aspect of context. Further, each contextual type can have a complicated structure reflecting complex nature of the contextual information. Although this complexity can take many different forms, one popular defining characteristic is the hierarchical structure of contextual information that can be represented as trees, as is done in most of the context-aware recommender and profiling systems [ASST05]. In the same way, it is possible to find OLAP-based representations widely used in the data warehousing applications.

Regarding the use of the contextual information in the recommendation process, the different approaches used can be broadly categorized in two groups: recommendation via context-driven querying and search, and recommendation via contextual preference elicitation and estimation. Context-driven querying and search basically uses the contextual information to query or search a certain repository of resources and suggest the best matching resources. In contrast to this approach, recommendation via contextual preference elicitation and estimation attempts to model and learn user preferences by observing the interactions of this and other users with the systems or by obtaining preference feedback from the user on various previously recommended items. To model users' context-sensitive preferences and generate recommendations, these techniques typically either adopt existing collaborative filtering, content-based, or hybrid recommendation methods to context-aware recommendation settings or apply various intelligent data analysis techniques from data mining or machine learning (such as Bayesian classifiers or support vector machines).

5.3 State-of-the-art for MICO use cases

5.3.1 Recommender systems in Citizen Science / Zooniverse Use Case

Related Technology Enablers: TE-502, TE-504, TE-505

The Zooniverse is a suite of online citizen science projects for which the general structure involved delivering a 'subject' (either image, video, or audio file) to a volunteer who is given the task of performing some sort of classification of the subject. For example, in Snapshot Serengeti volunteers will be delivered an image from a camera trap and asked to identify what species of animals they see in the image, how many of them there are, and what they are doing. Other task types found in Zooniverse projects include classification, comparison and annotation.

There are two main areas in which the implementation of recommender systems in the Zooniverse can be useful:

- *Cross-project recommendations*: With over twenty separate citizen science projects running on the Zooniverse platform at the moment (and many more under development) it would be advantageous to have a reliable way of recommending different projects to a specific volunteer or group of volunteers, perhaps to help out a project that does not have a high rate of classifications, or to retain a volunteer who we think is going to leave the project they are on.
- *Subject recommendations / task allocation*: Within a certain Zooniverse project it can be useful to deliver/recommend a certain subject (or type of subject) to a specific volunteer (or group of volunteers) based on some measurement of that volunteers' performance.

The following sections outline the state-of-the-art for recommender systems in the Zooniverse based on these two areas.

Cross-project recommendations
The Zooniverse currently hosts over twenty separate web-based citizen science projects, and it is adding new projects at a rate of about one every month. As projects have finite lifetimes, cross-project recommendations are essential in maintaining classifier activity; the majority of effort on most recent Zooniverse projects comes from existing users, and the majority of volunteers who sign up have tried more than a single project. At the moment, users on any given project can be given a recommendation to try another project in one of three very basic ways:

1. The top banner on the project they are using may show a recommendation to try another project. For example, on planethunters.org the top banner shows a statement that says *"Planet Hunters is part of the Zooniverse ...just like Moon Zoo"*. This recommendation is hard-wired by the Zooniverse developers and is not user-specific, i.e. the same recommendation is seen by all users on Planet Hunters. Many of the Zooniverse projects do not have this type of recommendation on their banner, and other projects have different recommendations.

2. The Zooniverse home page (`http://www.zooniverse.org/`), which presents the users with a list of all currently active projects, has a large top banner that displays a recommendation to try a specific project. Again this recommendation is hard-wired by the Zooniverse developers and is not user-specific. The banner is sometimes set to rotate through a list of projects that the Zooniverse would like to draw attention to for one reason or another. Other times it is fixed to one specific project for a longer period of time. More recently it has been fixed to one specific project but rotated through multiple versions with differing copy as an A/B split experiment to test user motivations and how to acquire more users into a certain project.

3. The Zooniverse provides a user-specific webpage (`http://www.zooniverse.org/me`) where individual users can see an overview of their own contributions across all projects. This includes a list of all projects in which that specific user has taken part. At the bottom of this webpage is a section entitled 'Your Recommendations' where four projects that the user has not contributed to are offered up as recommendations. The algorithm for choosing these four projects from the body of Zooniverse projects is basic. It presents the four most popular projects (as defined by how many other users are contributing to them) to which that specific user has not contributed already.

Subject recommendations / task allocation

Citizen science projects, in which subjects for classification or viewing are shown to a crowd of volunteers for processing, present an interesting recommendation problem. In addition to the usual aims of recommendation services – i.e. to supply something to a user which they would like to watch – there is an overarching desire to maximize the efficiency of the system. A user's desire to see the most interesting or engaging subjects is balanced by the need of the system to complete the task of sorting through all available subjects. In cases where we can assume that the users are motivated by task completion – for example, when participation in Zooniverse projects is motivated by a desire to help science ([RBG+13]) – the problem reduces to finding the most efficient set of task allocations.

The ideal task allocation system will maintain the accuracy of results while minimizing the total number of classifications required, without a noticeable delay in the time taken for task allocation. In practice, simple rules for subject retirement can be introduced; these are especially useful in situations in which a large number of subjects are not interesting. An example is found in *Snapshot Serengeti*, where the camera traps can be triggered not by a passing animal but by waving grass. When this happens, the rest of the card is usually filled with images of nothing but grass, but as identifying that nothing else is visible is an easy task, subjects in which the first three people report nothing are retired despite the fact they have been seen by rather few people. Similarly, in projects such as *Planet Hunters* where discoveries are rare (amongst a large number of non-interesting subjects) the system immediately follows up on potential discoveries by showing anything identified as a potential planet to the next available user.

A generic optimization algorithm for systems in which the choice is a binary one (as in *Planet Hunters* : Planet or Not-Planet) was developed by [KOS11] who show that for systems in which all tasks are the same cost (i.e. when it takes the same amount of effort to classify each subject) and

when workers are considered to be fleeting (i.e. when experience does not change over time) their approach is of order optimal, excluding a constant factor which depends on the specifics of the task and the population. By using random regular graphs for task allocation and a novel iterative algorithm to infer the correct answers, they are able to achieve good results even while maintaining a non-adaptive approach; that is, keeping the strategy the same regardless of the results received. This has a particular practical appeal, as a strategy could be calculated and implemented in advance of a project being launched. The task allocation algorithm first makes a choice of how many workers to assign to each task and how many tasks to give each worker, before worker reliability is assessed based on a post facto assessment of how and when they agree with the majority. One intriguing further result is that there exists a 'phase transition' around which the problem becomes complex enough for more sophisticated algorithms to begin to outperform simple majority voting; this suggests that for problems which are simple, or where user performance is consistent (experience suggests this will be true for the most simple *and* the most complex of crowdsourcing problems), there is little to be gained in investing processing time to make for a more complicated algorithm.

This approach, while easily generalizable and – subject to available processing time – implementable in a real-world crowdsourcing approach, throws away much information about the user. In platforms such as the Zooniverse, we have the ability to track users from task to task, and thus build up information about the likely performance of a user on a task. The assumption that the cost of each task is the same is also not always true; in *Snapshot Serengeti*, for example, images of elephants (which are easy to recognize) may have a lower cost than images of antelopes (for which the user is asked to distinguish between several species). In these circumstances, the provision of 'gold standard' data where the required result is already known, becomes important as a means of quickly determining user reliability. Such gold standard data might either be simulations (as in Zooniverse projects *Space Warps* or *Planet Hunters*) or be expert classified data.

Simpson et al. ([SRPS12]) developed a method of task assignment based on the use of such data. Their dataset was derived from the *Galaxy Zoo* : Supernova project, which characteristically involved the classification of a few thousand images via a simple decision tree. The individual answers to the decision tree were not considered, but the three possible exits resulted in three possible scores (confusingly, these are assigned values of -1, 1 or 3, but the categories and not their assigned values are used in Simpson's et al. analysis). Working in a Bayesian framework, as in the prior work of Ghahramani and Kim ([GK03], a confusion matrix is determined whose entries are the likelihood of a given answer given by any user to any of the available subjects. The advantages of this approach include natural handling of the sparseness of effort (most users do not see most subjects, especially in a project like this one where subjects were aggressively retired).

Ghahramani and Kim suggested the use of Gibbs Sampling ([GG84]), a form of inference that takes advantage of MCMC algorithms to investigate an underlying probability distribution, in order to take advantage of the resulting guarantee of accuracy, but this is often extremely slow to converge. As these codes, unlike those discussed above, are *adaptive* in that they take advantage of information provided in each classification, this is highly undesirable; users are unlikely to remain on the site while a recommendation is made. Instead, Simpson et al use a variational Bayes model in which sufficient statistics are deployed. For the example of archived *Galaxy Zoo: Supernova*, this approach not only outperformed majority voting or weighted voting but also substantially outperformed the (slower) Gibbs Sampling approach.

To make use of this approach as a recommendation engine, a further generalization is required in order to form a dynamic Bayesian classifier. A dynamic generalized linear model is used to iterate through the data to update the expected value for a task which has just been assigned, before a second pass updates all other expected values in turn. This produces an estimate of the optimal next task choice, but also gives each classifier a track record over time. Tasks can either be assigned to individual users, or to groups of users who have been identified to have similar behaviours; this latter tactic is most useful in situations such as that in citizen science where it cannot be assumed that a given classifier will remain on the site for a given amount of time.

This approach fails to consider the effect on a classifier's morale in its task assignments. In particular, it will in the ideal case produce in some cases a set of very similar tasks for each classifier, chosen according to the classifier's proficiency. Specifically, we might expect Galaxy Zoo classifiers to be assigned consistently bright, faint, spiral, elliptical, distant or nearby galaxies. If those classifiers are motivated by variety, or by a desire to see the most beautiful or spectacular images, then this will result in low volunteer retention rates. A model in which the cost of classification was allowed to vary from subject to subject, and in which the cost could be dynamically determined (for example, by measuring the rate of volunteer dropout after inspecting a given subject) would be hugely helpful in real world applications. A further extension, in which the model can not only assign test and gold standard data but also training data (in which the answer is both known and made visible to the volunteers) would also be of interest. In particular, one would want such a model to predict the likely effect of being trained on volunteer performance and longevity.

5.3.2 Recommender systems in news media / InsideOut10 Use Case

Online news reading has become extremely popular as the Internet enables access to news articles from millions of sources around the world. A key challenge of news websites as well as blogs is to drive traffic towards articles that readers find interesting to read.

News article recommendation differs in several ways from other types of recommender systems such as the ones designed for video or music even though at the basis we have content-based recommendation, user-based recommendation (or collaborative filtering) and hybrid approaches mixing the different methods.

The main differences between news recommendation when compared to music and videos recommendation systems are the following:

- freshness can be more important than relevancy,
- the unstructured format of a news story is more difficult to analyze than a music track with all its proper metadata,
- news readers might have an unpredictable preference for some particular events contained in news articles that is hard to discover other than by accessing personal information in the form of user actions (i.e. clickstreams on news sites, sharing or re-posting news articles over Facebook, posting photos on Flickr or video on YouTube in association with a specific event),
- news interests change over time and it is not always effective to use past-user actions to filter news content,

- the variety in recommended news articles (or blog post) is extremely valuable (serendipity effect) to drive traffic and engagement,
- breaking and trendy news articles might become of particular interest for a user even when not related to his or her general interests if they are being disseminated within his or her circle of friends over social networks (50% of social network users share or repost news stories, images or videos; nearly as many – 46% - discuss news issues or events on social network sites[64]) at the same time statistics show here that the virality of a piece of content (how much the article has been liked or shared over social networks) doesn't corresponds to an higher level of attention[65](meaning that the content shared is viewed but the quality of the engagement measured in terms of time spent or percentage of page scrolled is relatively low).

Despite the significant progress of recommender systems available today for publishers from mid-size companies like Zemanta and Taboola to large organizations like Google (planning this year to introduce a new recommendation platform for web publishers[66]) there are still major issues that limit the effectiveness of recommendation systems for news articles.

The following challenges are significant in the domain of news recommendation and somehow relevant in our use cases:

- cold-start users are hard to serve when they first request a recommendation as their interests remain unknown (user-based filtering); as the users interact with the content the recommendation system should be able to incrementally update the user profile reflecting his/her change in interest,
- cold-start news articles that have not yet been tied to many users' preferences (content-based filtering) are difficult to be repurposed (a problem also known as *first rater problem*, [DDGR07]),
- the user's preference for a given news article highly depends on his/her current context,
- consent to access personal information stored on social networks like Facebook, Google+ and Twitter is not granted by all users and in any case raises potential privacy issues,
- online news videos are becoming a core element of news articles and with broader mobile adoptions. These videos are in some cases gathered from citizens submitting online contents (using web forms or instant video recording applications like Shoof) or aggregated from social networks and blogs,
- mobile devices' screens have limited space available to user interfaces for recommendation.

Content-based recommendation repurpose items using either the lexical content of the previously viewed items or semantic information extracted with NLP tools. User based recommendation (or collaborative filtering) exploit profile similarities between different users.

While originally content-based recommendations were more common when filtering news content [AGHT11]) now hybrid approaches are widely used by large sites such as google news and yahoo and the recommendation heavily relies on the web history of logged-in users ([LCLS10])

[64] http://www.journalism.org/2014/03/26/8-key-takeaways-about-social-media-and-news/
[65] http://time.com/12933/what-you-think-you-know-about-the-web-is-wrong/
[66] http://venturebeat.com/2014/02/10/google-is-pushing-a-new-content-recommendation-system-for-publishers

User Profile
The user profile is a key part of efficient filtering in recommendation platforms. According to [BP99] there are two type of interest: short-term driven by the news agenda and its trend and long-term that most likely resembles the actual interests of a user. In order to provide valuable recommendation of news items a profile needs to be constructed and updated as change in interest is frequent especially for short-term interests.

The user interests can be determined based on the news items which have been read during a single navigation session or across multiple sessions with the use of web cookies or in the case of authenticated users. Other information can be derived from social profiles in the case of users that use the social login and provide their consent in accessing their social data. For semantics-based recommendation methods the concepts that appear in the news items are stored in vector and various algorithms can be used to assess concept equivalence and concept relatedness.

Considering the context of *Shoof* ("Look here" in arabic – a mobile instant video recording application for user generated content being developed by the Egyptian team of InsideOut10) and *WordLift* (a plugin for WordPress) at present stage, in terms of user profile, we have respectively access to the following:

- Shoof
  - click statistics for each browsing session of a user (e.g. which videos have been viewed, which categories, general site usage),
  - access to the twitter or facebook profile of the users (consent to access the social graph might be denied on login),
  - video content being uploaded, its meta-data (including geo-location) and the semantic data as a result of MICO analysis workflow,
- WordLift
  - click statistics for session,
  - WordPress profile for logged-in users and authors including links to their G+ profile. The wordpress profile consists in the following: username, first name, last name, nickname, email, short biography, website and (when available) list of contributed articles,
  - number of user comments for the article.

While for Shoof being a mobile application we can gather profiling data, for WordLift the ideal recommendation system will work using data gathered during each visit of anonymous web users. In other words for WordLift, as in most of news publishing platforms, a successful algorithm needs to degrade gracefully and be able to provide valuable recommendations even when there is little information about the user.

Articles and Media Recommendations in WordLift and Shoof
A recommendation system in the context of WordLift (news publishing websites and blogs) has to be fast and provide real-time recommendations without impacting on the overall performance of the website (page speed impacts the search ranking[67]). Flexibility is also important in order to respond properly to different classes of users with relevant recommendations. These classes of users including: anonymous

web users, anonymous web users being traced with the use of web cookies, profiled users (email only) and profiled users that are using a social login (Facebook, Twitter, G+ or WordPress). For Shoof the recommendation is primarily limited by the size of the screen that can be used for the re-purposing of content in-app. Users are in most cases profiled with either email or social login. In terms of user interactions the system shall be able to propose content in the form of videos (for Shoof and WordLift), text and images (WordLift only).

We envision the following dynamics for the user experience:

1. Contents to be pushed on page using interactive widgets being placed within the context of a news article and below the "fold"[68]– meaning in a relevant position exactly where the body of the content is. This specific position requires a general understanding of the text being written to be relevant for the user and not disturbing.

2. News articles or video to be promoted using web notifications[69] in desktop and mobile browsers that are starting to support this method[70].

3. For profiled users messaging can be used in the form of e-mail and sms (Shoof only) to suggest content that can be relevant for a user.

[67] http://www.mattcutts.com/blog/site-speed/
[68] http://timedotcom.files.wordpress.com/2014/03/unknown-1.png?w=560
[69] http://www.w3.org/TR/notifications/#examples
[70] http://caniuse.com/notifications

6 Related Implementations

Table 8 Component Sheet - FhG Temporal Video Segmentation

Related SotA section (name):	2.4.6
Implementation name:	FhG Temporal Video Segmentation
Description:	The temporal video segmentation software detects shots, key frames (i.e. important frames) and scenes in videos.
URL / source:	`http://www.idmt.fraunhofer.de`
Language:	CPP
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	executable, library
Current version:	1.3.0
Last Update:	23/04/2014
Software license:	commercial
Input/Output:	video/video/measurement values, images
Dependencies:	none
Comments:	none

Table 9 Component Sheet - Stanford NER

Related SotA section (name):	2.2, 2.3
Implementation name:	The Stanford named-entity extractor
Description:	A Java implementation of a named-entity extractor. The extractor finds entities such as persons, places, and names of organisations in plain text.
URL / source:	`http://nlp.stanford.edu/software/CRF-NER.shtml`
Language:	Java (JDK 1.6 or later)
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	Java package
Current version:	3.3.1
Last Update:	04/01/2014
Software license:	GNU General Public License (v2 or later)
Input/Output:	Text/annotations
Dependencies:	none
Comments:	none

Table 10 Component Sheet - The Stanford Parser

Related SotA section (name):	2.2, 2.3
Implementation name:	The Stanford Parser
Description:	A phrase structure and dependency parser. The parser analyses natural-language sentences and assigns them a syntactical structure.
URL / source:	`http://http://nlp.stanford.edu/software/lex-parser.shtml`
Language:	Java (JDK 1.6 or later)
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	Java package
Current version:	3.3.1
Last Update:	04/01/2014
Software license:	GNU General Public License (v2 or later)
Input/Output:	text/parse trees
Dependencies:	none
Comments:	none

Table 11 Component Sheet - Apache OpenNLP library

Related SotA section (name):	2.2, 2.3
Implementation name:	Apache OpenNLP library
Description:	Machine-learning toolkit for processing natural language. The toolkit can be used to train classifiers to detect sentiment, polarity, etc. of input documents.
URL / source:	`opennlp.apache.org`
Language:	Java (JDK 1.6 or later)
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	Library, source code
Current version:	1.5.3
Last Update:	24/04/2014
Software license:	Apache license, v. 2.0
Input/Output:	Training data, target document/Classifier, classification
Dependencies:	JDK 6, Apache Maven 3.0
Comments:	none

Table 12 Component Sheet - Freeling

Related SotA section (name):	2.2, 2.3
Implementation name:	Freeling
Description:	An open source suite of natural language analyzers to handle a range of analysis tasks
URL / source:	`http://nlp.lsi.upc.edu/freeling/`
Language:	C++, Java bindings via JNI
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	Library, source code
Current version:	3.1
Last Update:	13/09/2013
Software license:	GNU General Public License, v. 3.0
Input/Output:	Text/Annotations
Dependencies:	see Documentation
Comments:	none

Table 13 Component Sheet - Apache Stanbol

Related SotA section (name):	2.2
Implementation name:	Apache Stanbol
Description:	An open source suite for semantic content management
URL / source:	`http://stanbol.apache.org`
Language:	Java
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	RESTful API, Server
Current version:	0.12
Last Update:	02/03/2014
Software license:	Apache license, v. 2.0
Input/Output:	Text, Rich Text/Detected Entities, Extracted Entities, Categories,
Dependencies:	see Documentation
Comments:	Stanbol integrates with other NLP frameworks such as OpenNLP, Stanford NLP and Freeling. It allows link Entities from user managed vocabularies

Table 14 Component Sheet - CMU Sphinx

Related SotA section (name):	2.2
Implementation name:	CMU Sphinx
Description:	Open source toolkit for automatic speech recognition
URL / source:	`http://cmusphinx.sourceforge.net`
Language:	Java (JDK 1.6 or later)
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	Library, source code
Current version:	Sphinx4, v. 1.0 Beta 6
Last Update:	03/01/2011
Software license:	Apache license v.2
Input/Output:	Sound and language models/time-stamped transcriptions
Dependencies:	none
Comments:	none

Table 15 Component Sheet - Python Natural Language Toolkit

Related SotA section (name):	2.2, 2.3
Implementation name:	Python NLTK
Description:	Platform for building Python programs to work with natural language.
URL / source:	`http://www.nltk.org`
Language:	Python versions 2.6-2.7, 3
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	Library, source code
Current version:	3.0
Last Update:	01/11/2013
Software license:	BSD-style license
Input/Output:	Text/annotations
Dependencies:	none
Comments:	none

Table 16 Component Sheet - OpenIMAJ

Related SotA section (name):	2.4.1 , 2.4.2, 2.4.4, 2.4.6
Implementation name:	OpenIMAJ
Description:	multimedia content analysis and content generation
URL / source:	`http://www.openimaj.org/`
Language:	Java
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	library
Current version:	1.2.1
Last Update:	09/03/2014
Software license:	BSD style license
Input/Output:	image, video/java api, output images and videos
Dependencies:	Java 7
Comments:	none

Table 17 Component Sheet - SHORE

Related SotA section (name):	*2.4.2,2.4.4*
Implementation name:	*SHORE^{TM}*
Description:	*face detection and facial analysis of faces for humans and great apes [KE06]*
URL / source:	*`http://www.iis.fraunhofer.de`*
Language:	*CPP*
Platform:	*Windows (32/64), Linux (32/64)*
type:	*executable, library*
Current version:	*140*
Last Update:	*08/04/2014*
Software license:	*commercial*
Input/Output:	*input: image, video, output: annotations*
Dependencies:	—-
Comments:	—-

Table 18 Component Sheet - OpenCV

Related SotA section (name):	*2.4.2,2.4.4,2.4.1*
Implementation name:	*OpenCV*
Description:	*face detection[VJ01], various low-level features, face recognition [TP91b]*
URL / source:	`http://opencv.org/`
Language:	*CPP*
Platform:	*Windows (32/64), Linux (32/64), OSX, Android, iOS*
type:	*libraries*
Current version:	*2.4.9*
Last Update:	*25/04/2014*
Software license:	*BSD*
Input/Output:	*input: image, video, output: annotations, images, features*
Dependencies:	—-
Comments:	—-

Table 19 Component Sheet - VLFeat

Related SotA section (name):	*2.4.2,2.4.4,2.4.1*
Implementation name:	*VLFeat*
Description:	*various low-level features, detectors and CV algorithms*
URL / source:	`http://www.vlfeat.org/`
Language:	*CPP, MATLAB*
Platform:	*Windows (32/64), Linux (32/64), OSX*
type:	*libraries, source code*
Current version:	*0.9.18*
Last Update:	*29/01/2014*
Software license:	*BSD*
Input/Output:	*input: image, video, output: annotations, images, features*
Dependencies:	—-
Comments:	—-

Table 20 Component Sheet - FhG software modules

Related SotA section (name):	*2.4.1*
Implementation name:	*FhG software modules*
Description:	*various low-level features*
URL / source:	`http://idmt.fraunhofer.de/`
Language:	*CPP, MATLAB*
Platform:	*Windows (32/64), Linux (32/64), OSX*
type:	*shared libraries, m-files*
Current version:	—
Last Update:	*25/04/2014*
Software license:	*closed source*
Input/Output:	*input: image, video, output: features*
Dependencies:	*VLFeat, OpenCV, Intel IPP*
Comments:	—-

Table 21 Component Sheet - LIBSVM

Related SotA section (name):	*2.4.2,2.4.4,2.4.1*
Implementation name:	*LIBSVM*
Description:	*library for classification using Support Vector Machines*
URL / source:	`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
Language:	*CPP, MATLAB-interface, Python-interface, Java-Interface*
Platform:	*Windows (32/64), Linux (32/64), OSX*
type:	*libraries*
Current version:	*3.18*
Last Update:	*01/04/2014*
Software license:	*BSD license*
Input/Output:	*input: feature vectors, output: classifications, confidences*
Dependencies:	—-
Comments:	—-

Table 22 Component Sheet - FhG AFR

Related SotA section (name):	*2.4.4*
Implementation name:	*FhG AFR software modules*
Description:	*libraries, source code for SOTA face recognition approaches [TP91b, BHK97, He05, WYG+09, YZY11, YCCY13, YZ10, Eke09, AHP06, ZSG+05]*
URL / source:	`http://www.idmt.fraunhofer.de`
Language:	*CPP, MATLAB*
Platform:	*Windows (32/64), Linux (32/64), OSX*
type:	*libraries, m-files*
Current version:	—
Last Update:	*25/04/2014*
Software license:	*closed source*
Input/Output:	*input: facial images output: classification results*
Dependencies:	—-
Comments:	—-

Table 23 Component Sheet - FhG software modules for dimensionality reduction

Related SotA section (name):	*2.4.2,2.4.4*
Implementation name:	*FhG software modules for dimensionality reduction*
Description:	*libraries, source code for dimensionality reduction techniques (PCA, LDA, LPP, ...)*
URL / source:	`http://www.idmt.fraunhofer.de`
Language:	*CPP, MATLAB*
Platform:	*Windows (32/64), Linux (32/64), OSX*
type:	*libraries, m-files*
Current version:	—
Last Update:	*25/04/2014*
Software license:	*closed source*
Input/Output:	*input: high-dimensional feature vectors, output: low-dimensional feature vectors*
Dependencies:	—-
Comments:	—-

Table 24 Component Sheet - FhG software modules for object/animal detection and recognition

Related SotA section (name):	*2.4.2,2.4.3*
Implementation name:	*FhG software modules for object/animal detection and recognition*
Description:	*source code for object/animal detection and recognition in images and videos*
URL / source:	*http://www.idmt.fraunhofer.de*
Language:	*MATLAB*
Platform:	*Windows (32/64), Linux (32/64), OSX*
type:	*m-files*
Current version:	—
Last Update:	*25/04/2014*
Software license:	*closed source*
Input/Output:	*input: images, image sequences, output: detection and classification results*
Dependencies:	—-
Comments:	—-

Table 25 Component Sheet - FhG BEMVisual and BEMAudio and MediaQuality

Related SotA section (name):	2.4.5
Implementation name:	FhG BEMVisual and BEMAudio and MediaQuality
Description:	The Broadcast Error Monitoring (BEM) software detects audio and visual errors. Media quality allows cross-medial combining of measurement values.
URL / source:	http://www.idmt.fraunhofer.de
Language:	CPP
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	executable, library
Current version:	1.3.0
Last Update:	23/04/2014
Software license:	commercial
Input/Output:	audio+video/measurement values, images
Dependencies:	none
Comments:	none

Table 26 Component Sheet - Interra Systems - Baton

Related SotA section (name):	2.4.5
Implementation name:	Interra Systems - Baton
Description:	File-based QC and content verification
URL / source:	`http://www.interrasystems.com/file-based-qc.php`
Language:	unknown
Platform:	Windows, Linux
type:	software/executable (GUI)
Current version:	unknown
Last Update:	unknown
Software license:	commercial
Input/Output:	audio+video/GUI output, html-, xml-, pdf-report
Dependencies:	unknown
Comments:	unknown

Table 27 Component Sheet - R&S VEGA Suite

Related SotA section (name):	2.4.5
Implementation name:	R&S VEGA Suite
Description:	File-based QC
URL / source:	`http://www.rohde-schwarz.com/en/product/vega-productstartpage_63493-11300.html`
Language:	unknown
Platform:	Windows XP / 7 (32/64 bit)
type:	software/executable (GUI)
Current version:	unknown
Last Update:	unknown
Software license:	commercial
Input/Output:	audio+video files, MXF streams/GUI output
Dependencies:	unknown
Comments:	unknown

Table 28 Component Sheet - ShotDetect

Related SotA section (name):	2.4.6
Implementation name:	ShotDetect
Description:	ShotDetect is a free software which detects shots and scenes from a video.
URL / source:	`http://johmathe.name/shotdetect.html`
Language:	CPP
Platform:	Windows, Linux
type:	source code
Current version:	1.0.85
Last Update:	unknown
Software license:	LGPL
Input/Output:	video/xml-report
Dependencies:	none
Comments:	none

Table 29 Component Sheet - FhG XPX - Audio/Visual Low-Lovel Feature Extraction

Related SotA section (name):	2.4.1
Implementation name:	FhG XPX - Audio/Visual Low-Level Feature Extraction
Description:	Extraction of audio and visual low-level features.
URL / source:	`http://www.idmt.fraunhofer.de`
Language:	CPP
Platform:	Windows, Linux, OS X
type:	API, executable (CLI)
Current version:	1.0
Last Update:	unknown
Software license:	commercial
Input/Output:	audio+video/ low-level features
Dependencies:	none
Comments:	none

Table 30 Component Sheet - CVLAB - DAISY: A Fast Local Descriptor for Dense Matching

Related SotA section (name):	2.4.1
Implementation name:	CVLAB - DAISY: A Fast Local Descriptor for Dense Matching
Description:	An Efficient Dense Descriptor Applied for Wide Baseline Stereo.
URL / source:	`http://cvlab.epfl.ch/software/daisy`
Language:	CPP
Platform:	Windows, Linux, OS X
type:	Matlab / CPP Source Code
Current version:	1.8.1 (CPP), 1.0 (Matlab)
Last Update:	11/10/2009
Software license:	BSD
Input/Output:	images/ visualization
Dependencies:	none
Comments:	none

Table 31 Component Sheet - FhG Music-Video Annotation Tool

Related SotA section (name):	2.4.8
Implementation name:	FhG Music-Video Annotation Tool
Description:	Annotation spec and UI supporting the manual annotation of semantic concepts
URL / source:	`http:/www.idmt.fraunhofer.de`
Language:	CPP
Platform:	Windows (32/64), Linux (32/64), OS X
type:	CPP Source Code
Current version:	
Last Update:	28/04/2014
Software license:	closed source
Input/Output:	audio, video, images / annotations
Dependencies:	none
Comments:	none

Table 32 Component Sheet - Queen Mary University Sonic Visualiser

Related SotA section (name):	2.4.8
Implementation name:	Sonic Visualiser
Description:	Application for viewing, analysing, and annotating of audio files
URL / source:	`http://www.sonicvisualiser.org/`
Language:	CPP
Platform:	Windows (32/64), Linux (32/64), OS X
type:	CPP Source Code
Current version:	
Last Update:	13/12/2013
Software license:	GNU General Public License (v2 or later)
Input/Output:	audio / features, annotations
Dependencies:	none
Comments:	none

Table 33 Component Sheet - FhG Soundslike

Related SotA section (name):	2.4.9
Implementation name:	FhG Soundslike
Description:	Music similarity search
URL / source:	`http://www.idmt.fraunhofer.de/`
Language:	CPP
Platform:	Windows (32/64), Linux (32/64), OS X
type:	CPP Source Code
Current version:	
Last Update:	28/04/2014
Software license:	closed source
Input/Output:	audio / similarity results
Dependencies:	none
Comments:	none

Table 34 Component Sheet - FhG Music Annotation

Related SotA section (name):	2.4.8
Implementation name:	FhG Music Annotation
Description:	Annotation of various musical properties and concepts
URL / source:	`http://www.idmt.fraunhofer.de/`
Language:	CPP
Platform:	Windows (32/64), Linux (32/64), OS X
type:	CPP Source Code
Current version:	
Last Update:	28/04/2014
Software license:	closed source
Input/Output:	audio / features, annotations, concepts
Dependencies:	none
Comments:	none

Table 35 Component Sheet - FhG Speech-Music Discrimination

Related SotA section (name):	2.4.7
Implementation name:	FhG Speech-Music Discrimination
Description:	Discriminate segments which contain music vs. speech vs. other
URL / source:	`http://www.idmt.fraunhofer.de/`
Language:	CPP
Platform:	Windows (32/64), Linux (32/64), OS X
type:	CPP Source Code
Current version:	
Last Update:	28/04/2014
Software license:	closed source
Input/Output:	audio / segmentation
Dependencies:	none
Comments:	none

Table 36 Component Sheet - SPARQL 1.1

Related SotA section (name):	4.2.1
Implementation name:	SPARQL 1.1
Description:	Marmotta SPARQL module provides support for SPARQL 1.1
URL / source:	`https://marmotta.apache.org`
Language:	Java
Platform:	Windows (32/64), Linux (32/64), OSX(64)
type:	plattform module
Current version:	3.2.0
Last Update:	25/04/2014
Software license:	BSD style license
Input/Output:	sparql requests and responses in various formats
Dependencies:	Java 7
Comments:	none

This is a list of planned implementations. There is no guarantee that all implementations will be considered for release within the MICO platform. The implementations will be developed on Use Case based priorities. At the editorial deadline these priorities were not yet fixed. Information about the current state of implementations can be found in the release reports.

References

[AAA+11] Mongi Abidi, Jörgen Ahlberg, Brian Amberg, Heinrich H. Bülthoff, Simon Baker, Ronen Basri, Wei Bian, Volker Blanz, Michael Brauckmann, Christoph Busch, Hong Chang, Rama Chellappa, D. Chu, Jeffrey F. Cohn, Tim Cootes, Douglas W. Cunningham, Xiaoqing Ding, Ming Du, E. Efraty, Ralph Gross, Patrick Grother, Baining Guo, Abdenour Hadid, Thomas Huang, David Jacobs, Anil K. Jain, I.A. Kakadiaris, Takeo Kanade, Reinhard Knothe, Andreas Koschan, Joni-Kristian Kämäräinen, Stan Z. Li, Xiaoming Liu, Zicheng Liu, J. Birgitta Martinkauppi, Iain Matthwes, Ross Micheals, Baback Moghaddam, Alice J. O'Toole, Igor S. Pandzic, Sharathchandra Pankanti, Usang Park, G. Passalis, and P. Perakis. *Handbook of Face Recognition*. Springer London Limited 2011, London, second edi edition, 2011.

[AG07] Donovan Artz and Yolanda Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71, 2007.

[AGHT11] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing user modeling on twitter for personalized news recommendations. In *User Modeling, Adaption and Personalization*, pages 1–12. Springer, 2011.

[AH04] Timo Ahonen and Abdenour Hadid. Face recognition with local binary patterns. In *European Conference on Computer Vision (ECCV)*, pages 469–481, 2004.

[AHP06] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(12):2037–2041, 2006.

[AKB08] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In *Computer Vision ECCV 2008*, volume 5305, pages 102–115, 2008.

[All83] James F. Allen. Maintaining knowledge about temporal intervals, 1983.

[AMCG04] Florina Almenárez, Andrés Marín, Celeste Campo, and Carlos Garcia. Ptm: A pervasive trust management model for dynamic open environments. In *First Workshop on Pervasive Security, Privacy and Trust PSPT*, volume 4, pages 1–8, 2004.

[AMK99] S. Abbasi, F. Mokhtarian, and J. Kittler. Curvature scale space image in shape similarity retrieval. *Multimedia Systems*, 7(6):467–476, 1999.

[Ang87] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

[AOV12] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, Piscataway, 2012. Ieee.

[ASST05] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, January 2005.

[ASSW01] Tom Arbuckle, Stefan Schröder, Volker Steinhage, and Dieter Wittmann. Biodiversity Informatics in Action: Identification and Monitoring of Bee Species using ABIS. In *International Symposium on Informatics for Environmental Protection (EnviroInfo)*, pages 425–430, Zurich, Switzerland, 2001.

[ATEP08] Heydar Maboudi Afkham, Alireza Tavakoli Targhi, Jan-Olof Eklundh, and Andrzej Pronobis. Joint Visual Vocabulary for Animal Classification. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4, Tamper, Florida, USA, December 2008.

[Bay07] David M Baylon. On the detection of temporal field order in interlaced video data. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 6, pages VI–129. IEEE, 2007.

[BBCZ05] G.B. Bezerra, T.V. Barra, L.N. Castro, and F.J.V. Zuben. Adaptive radius immune algorithm for data clustering. *Artificial Immune Systems, LNCS*, 3627:290–303, 2005.

[BBD10] Suna Bensch, Henrik Björklund, and Frank Drewes. Algorithmic properties of Millstream systems. In Y. Gao, H. Lu, S. Seki, and S. Yu, editors, *Proc. 14th International Conference on Developments in Language Theory (DLT 2010)*, volume 6224 of *Lecture Notes in Computer Science*, pages 54–65. Springer, 2010.

[BC06] Tilo Burghardt and Janko Calic. Real-Time Face Detection and Tracking of Animals. In *Seminar on Neural Network Applications in Electrical Engineering (NEUREL)*, pages 27–32, Belgrade, Serbia, 2006.

[BC10] Tilo Burghardt and Neill Campbell. Generic Phase Curl Localisation for Individual Identification of Turing-Patterned Animals. In *Workshop on Visual Observation and Analysis of Animal and Insect Behavior (VAIB)*, pages 17–21, Istanbul, Turkey, 2010.

[BCD05] Roberto Basili, Marco Cammisa, and Emanuale Donati. Ritroverai: A web application for semantic indexing and hyperlinking of multimedia news. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *Proc. 4th International Semantic Web Conference (ISWC 2005)*, volume 3729 of *Lecture Notes in Computer Science*. Springer, 2005.

[BCF+07] Scott Boag, Don Chamberlin, Mary F Fernández, Daniela Florescu, Jonathan Robie, and Jérôme Siméon. XQuery 1.0: An XML Query Language. Technical report, 2007.

[BCM+05] Jesús Bescós, Guillermo Cisneros, José M Martínez, José M Menéndez, and Julián Cabrera. A unified model for techniques on video-shot transition detection. *Multimedia, IEEE Transactions on*, 7(2):293–307, 2005.

[BCT04] Tilo Burghardt, Janko Calic, and Barry T. Thomas. Tracking Animals in Wildlife Videos Using Face Detection. In *European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, London, UK, 2004.

[BDJvdM14] Suna Bensch, Frank Drewes, Helmut Jürgensen, and Brink van der Merwe. Graph transformation for incremental natural language analysis. *Theoretical Computer Science*, 531:1–25, 2014.

[BETVG08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[BF06] Tamara L. Berg and David A. Forsyth. Animals on the Web. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1463–1470, New York, USA, 2006.

[BFG01] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual web information extraction with Lixto. In Peter M. G. Apers, Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Kotagiri Ramamohanarao, and Richard Thomas Snodgrass, editors, *Proc. 27th International Conference on Very Large Data Bases (VLDB'01)*, pages 119–128. Morgan Kaufmann Publishers, 2001.

[BFHT09] Kofi Boakye, Benoît Favre, and Dilek Hakkani-Tür. Any questions? automatic question detection in meetings. In *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU 2009)*, pages 485–489. IEEE Computer Society, 2009.

[BFS05] James Bailey, Tim Furche, and Sebastian Schaffert. Web and Semantic Web Query Languages : A Survey. In *Reasoning Web*, pages 35–133, 2005.

[BG09] G.J. Burghouts and J.M. Geusebroek. Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, 113(1):48–62, 2009.

[BHBL09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[BHK97] Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis andPattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[BL06] Tim Berners-Lee. Linked Data. `http://www.w3.org/DesignIssues/LinkedData.html`, 2006.

[BM11] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[BNJ03] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[BP99] Daniel Billsus and Michael J Pazzani. A hybrid user model for news story classification. *COURSES AND LECTURES-INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES*, 99:108, 1999.

[BPVdWK06] Ian S Burnett, Fernando Pereira, Rik Van de Walle, and Rob Koenen. *The MPEG-21 book*. Wiley Online Library, 2006.

[Bre01] L. Breiman. Randomforest. *Machine Learning*, 45:5–32, 2001.

[BS05] Rémi Barland and Abdelhakim Saadane. Reference free quality metric for jpeg-2000 compressed images. In *Signal Processing and Its Applications, 2005. Proceedings of the Eighth International Symposium on*, volume 1, pages 351–354. IEEE, 2005.

[BSW+11] Dimitry Bogdanov, Joan Serra, N. Wack, Perfecto Herrera, and X. Serra. Unifying low-level and high-level music similarity measures. *IEEE Transactions on Multimedia*, 13(4):687–701, 2011.

[BTG06] Herbert Bay, Tinne Tuytelaars, and Luc Gool. Surf: Speeded up robust features. In *Computer Vision ECCV 2006*, volume 3951, pages 404–417, 2006.

[Bur08] Tilo Burghardt. *Visual Animal Biometrics - Automatic Detection and Individual Identification by Coat Pattern*. Phd thesis, University of Bristol, 2008.

[BV01] Y. Boykov and R. Veksler, H.and Zabih. Fast approximate energy minimization via graph cuts. *IEEE Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[BvD11] O. Barnich and M. van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, 2011.

[CBB+00] R. G. Cattell, Douglas K. Barry, Mark Berler, Jeff Eastman, David Jordan, Conn Russell, Olaf Schadow, Torsten Stanienda, and Fernando Velez. *The Object Data Standard: ODMG 3.0*. Morgan Kaufmann, the morgan edition, 2000.

[CCJP99] T. Choudhry, B. Clarkson, T. Jebara, and A. Pentland. Multimodal person recognition using unconstrained audio and video. In *International Conference on Audio and Video-Based Person Authentication*, 1999.

[CDF+04] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[Cel08] Òscar Celma. *Music recommendation and discovery in the long tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona and Spain, 2008.

[CGLN07] Julien Carme, Rémi Gilleron, Aurélien Lemay, and Joachim Niehren. Interactive learning of node selecting tree transducer. *Machine Learning*, 66(1):33–67, 2007.

[CGM02] C. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *International Conference on Pattern Recognition (ICPR)*, pages 150–155, Quebec, Canada, 2002.

[CGP05] Costas Cotsaces, M Gavrielides, and Ioannis Pitas. A survey of recent work in video shot boundary detection. In *Proceedings of 2005 Workshop on Audio-Visual Content and Information Visualization in Digital Libraries, available at: http://poseidon. csd. auth. gr/papers/PUBLISHED/CONFERENCE/pdf/Cotsaces05a. pdf*, 2005.

[Cho56] N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956.

[Cho06] Jinsook Cho. The mechanism of trust and distrust formation and their relational outcomes. *Journal of retailing*, 82(1):25–35, 2006.

[CLHA08] Yuchou Chang, Dah-Jye Lee, Yi Hong, and James Archibald. Unsupervised video shot detection using clustering ensemble with a color global scale-invariant feature transform descriptor. *Journal on Image and Video Processing*, 2008:9, 2008.

[CLMW11] E.J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011.

[CLSF10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Computer Vision – ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 778–792, Berlin and Heidelberg, 2010. Springer Berlin Heidelberg.

[CMY10] Jingwei Cheng, Z.M. Ma, and Li Yan. f-sparql: A flexible extension of sparql. In PabloGarcía Bringas, Abdelkader Hameurlain, and Gerald Quirchmayr, editors, *Database and Expert Systems Applications*, volume 6261 of *Lecture Notes in Computer Science*, pages 487–494. Springer Berlin Heidelberg, 2010.

[CNT04] Julien Carme, Joachim Niehren, and Marc Tommasi. Querying unranked trees with stepwise tree automata. In Vincent van Oostrom, editor, *Proc. 19th International Conference on Rewriting Techniques and Applications*, volume 3091 of *Lecture Notes in Computer Science*, pages 105–118. Springer, 2004.

[Cox95] G.S. Cox. Template matching and measures of match in image processing. Technical report, Department of Electrical Engineering, University of Cape Town, 1995.

[CPLT99] Michael J. Carey, Eluned S. Paris, and Harvey Lloyd-Thomas. A comparison of features for speech, music discrimination. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 149–152, 1999.

[CS08] Li Chen and F.W.M Stentiford. Video sequence matching based on temporal ordinal measurement. *Pattern Recognition Letters*, 29(13):1824–1831, 2008.

[CS14] Christian Chiarcos and Maria Sukhareva. OLiA - ontologies of linguistic annotation. *Semantic Web – Interoperability, Usability, Applicability*, 2014. To appear.

[CSM06] Baris Coskun, Bulent Sankur, and Nasir Memon. Spatio-temporal transform based video hashing. *IEEE Transactions on Multimedia*, 8(6):1190–1208, 2006.

[CSRK11] Shyamprasad Chikkerur, Vijay Sundaram, Martin Reisslein, and Lina J Karam. Objective video quality assessment methods: A classification, review, and performance comparison. *Broadcasting, IEEE Transactions on*, 57(2):165–182, 2011.

[CWL+08] Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. Finding question-answer pairs from online forums. In *Proc. 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*, pages 467–474. ACM, 2008.

[CWS95] R. Chellappa, C.L. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83:705–740, 1995.

[Cyg05] Richard Cyganiak. A relational algebra for SPARQL. Technical report, 2005.

[CZ03] Sen-ching Samson Cheung and Avideh Zakhor. Efficient video similarity measurement with video signature. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1):59–74, 2003.

[DC96] J D N Dionisio and A F Cardenas. MQuery: A visual query language for multimedia, timeline and simulation data. *J Visual Languages and Computing*, 7:377–401, 1996.

[DC01] Sanjiv Das and Mike Chen. Yahoo! for amazon: Extracting market sentiment from stock message boards. 2001.

[dCvZ01] L. de Castro and F. von Zuben. ainet: An artificial immune network for data analysis. *Data Mining: A Heuristic Approach*, pages 231–259, 2001.

[DDGR07] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, 2007.

[Del09] Jean-Yves Delort. Automatically characterizing salience using readers' feedback. *Journal of Digital Information*, 10(1), 2009.

[DH03] Frank Drewes and Johanna Högberg. Learning a regular tree language from a teacher. In Z. Ésik and Z. Fülöp, editors, *Proc. 7th International Conference on Developments in Language Theory (DLT'03)*, volume 2710 of *Lecture Notes in Computer Science*, pages 279–291. Springer, 2003.

[DH09] Dima Damen and David Hogg. Attribute multiset grammars for global explanations of activities. In Andrea Cavallaro, Simon Prince, and Daniel C. Alexander, editors, *Proc. British Machine Vision Conference (BMVC 2009)*, pages 1–11. British Machine Vision Association, 2009.

[DH12] Dima Damen and David Hogg. Explaining activities as consistent groups of events - a bayesian framework using attribute multiset grammars. *International Journal of Computer Vision*, 98:83–102, 2012.

[DHN99] M. Do, J. Harp, and K. Norris. A test of a pattern recognition system for identification of spiders. *Bulletin of Entomological Research*, 89:217–224, 1999.

[DKN08] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11(2):77–107, 2008.

[DLH05] Colin De La Higuera. A bibliographical study of grammatical inference. *Pattern recognition*, 38(9):1332–1348, 2005.

[DM07] Dipanjan Das and André F.T. Martins. A survey on automatic text summarization. Literature survey for the course Language and Statistics II. Technical report, Carnegie Mellon University, 2007.

[dMMM06] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parsers from phrase strucutre parses. In *Proc. 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 449–454, 2006.

[DP11] Kyle D. Dent and Sharoda A. Paul. Through the twitter glass: Detecting questions in micro-text. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence – Analyzing Microtext (WS-11-05)*. The AAAI Press, 2011.

[Dre09] Frank Drewes. MAT learners for recognizable tree languages and tree series. *Acta Cybernetica*, 19(2):249–274, 2009.

[DSB+05] Stefan Decker, Michael Sintek, Andreas Billig, Nicola Henze, Peter Dolog, Wolfgang Nejdl, Andreas Harth, Andreas Leicher, Susanne Busse, José Luis Ambite, Matthew Weathers, Gustaf Neumann, and Uwe Zdun. TRIPLE - an RDF Rule Language with Context and Use Cases. In *Rule Languages for Interoperability*, 2005.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, 2005.

[DTG+08] M. Döller, R. Tous, M. Gruhne, K. Yoon, M. Sano, and I.S. Burnett. The MPEG Query Format: Unifying Access to Multimedia Retrieval Systems. *IEEE Multimedia*, 15, 2008.

[DTR10] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proc. 14th Conference on Computational Natural Language Learning (CoNLL'10)*, pages 107–116. The Association for Computational Linguistics, 2010.

[dZPR+10] P.M. de Zeeuw, E.J. Pauwels, E.B. Ranguelova, D.M. Buonantony, and S.A. Eckert. Computer Assisted Photo Identification of Dermochelys Coriacea. In *Visual Observation and Analysis of Animal and Insect Behavior (VAIB)*, pages 165–172, Istanbul, Turkey, 2010.

[ECD+06] Duane R. Edgington, Danelle E. Cline, Daniel Davis, Ishbel Kerkez, and Jerome Mariette. Detection, Tracking and Classifying Animals in Underwater Video. In *OCEANS*, pages 1 – 5, Boston, Massachusetts, USA, 2006.

[EHD00] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision (ECCV)*, pages 751–767, Dublin, Ireland, 2000.

[Eid03] Horst Eidenberger. How good are the visual mpeg-7 features? In *Visual Communications and Image Processing 2003*, SPIE Proceedings, pages 476–488. Spie, 2003.

[EK11] A. Ernst and C. Küblbeck. Fast Face Detection and Species Cassification of African Great Apes. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 279–284, Klagenfurt, Austria, August 2011. IEEE.

[Eke09] Hazim Kemal Ekenel. *A Robust Face Recognition Algorithm for Real-World Applications*. PhD thesis, Universitaet Fridericiana zu Karlsruhe (TH), 2009.

[EMC09] Mosalam Ebrahimi and Walterio W. Mayol-Cuevas. Susure: Speeded up surround extrema feature detector and descriptor for realtime applications. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 9–14, 2009.

[ERK11] Michael D Ekstrand, John T Riedl, and Joseph A Konstan. *Collaborative filtering recommender systems*. Now Publishers Inc, 2011.

[ES05] Hazim Kemal Ekenel and Rainer Stiefelhagen. Local Appearance Based Face Recognition Using Discrete Cosine Transform. In *European Signal Processing Conference (EUSIPCO)*, 2005.

[ETC98] G.J. Edwards, C.J. Taylor, and T.F. Cootes. Learning to identify and track faces in image sequences. In *International Conference on AutomaticFace and Gesture Recognition*, 1998.

[EVGW+10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes chal- lenge 2010 (voc2010) results. Technical report, 2010.

[EWLH02] M.J. Er, S. Wu, J. Lu, and L-T. Hock. Face recognition with radial basis function (rbf) neural networks. *IEEE Transactions on Neural Networks*, 13(3):697–710, 2002.

[FBF+94] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3-4):231–262, 1994.

[FC09] Yu-Cheng Fan and Chia-Hao Chung. De-interlacing algorithm using spatial-temporal correlation-assisted motion estimation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(7):932–944, 2009.

[FGMR10] P.F. Felzenszwalb, R.B. Grishick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[FM05] Mylene CQ Farias and Sanjit K Mitra. No-reference video quality metric based on artifact measurements. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–141. IEEE, 2005.

[FPC03] Rino Falcone, Giovanni Pezzulo, and Cristiano Castelfranchi. A fuzzy approach to a belief-based trust computation. In *Trust, reputation, and security: theories and practice*, pages 73–86. Springer, 2003.

[FS99] Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 15(5):771–780, 1999.

[Fuk95] Francis Fukuyama. *Trust: The social virtues and the creation of prosperity*. Free Press New York, 1995.

[FWCT13] Lee Feigenbaum, Gregory Todd Williams, Kendall Grant Clark, and Elias Torres. SPARQL 1.1 Protocol, 2013.

[FWX09] Zhong-Hua Fu, Jhing-Fa Wang, and Lei Xie. Noise robust features for speech/music discrimination in real-time telecommunication. In *2009 IEEE International Conference on Multimedia and Expo (ICME)*, pages 574–577, 2009.

[Gar04] C. Garcia. A survey of face detection and recognition techniques. Technical report, France Telecom R&D, 2004.

[GC07] Costantino Grana and Rita Cucchiara. Linear transition detection as a unified shot detection approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(4):483, 2007.

[GG84] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.

[GK03] Zoubin Ghahramani and Hyun-Chul Kim. Bayesian classifier combination. 2003.

[GKRT04] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM, 2004.

[GMP00] Shaogang Gong, Stephen J. McKenna, and Alexandra Psarrou. *Dynamic Vision: From Images to Face Recognition*. Imperial College Press, London, UK, 2000.

[GMTT06] Rémi Gilleron, Patrick Marty, Marc Tommasi, and Fabien Torre. Interactive tuples extraction from semi-structured data. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006)*, pages 997–1004. IEEE Computer Society, 2006.

[GO04] Kevin J. Gaston and Mark A. O'Neill. Automated Species Identification: Why Not? *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 359(1444):655–67, April 2004.

[GPH03] Jennifer Golbeck, Bijan Parsia, and James Hendler. *Trust networks on the semantic web*. Springer, 2003.

[GPP13] Paul Gearon, Alexandre Passant, and Axel Polleres. SPARQL 1.1 Update, 2013.

[GS13] Alberto Gil Solla and Rafael G. Sotelo Bovino. *TV-Anytime*. X.media.publishing. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Guh03] Ramanathan Guha. Open rating systems. *URL:¡ citeseer. ist. psu. edu/694373. html*, 2003.

[GZT10] S. Gu, Y. Zheng, and C. Tomasi. Efficient visual object tracking with online nearest neighbor classifier. In *Asian Conference on Computer Vision (ACCV)*, 2010.

[Haj98] Jan Hajič. Building a syntactically annotated corpus: The prague dependency treebank. In Eva Hajičová, editor, *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová*, pages 12–19. Prague Karolinum, Charles University Press, 1998.

[Ham09] Abdul Hameed. Video shot detection by motion estimation and compensation. In *Emerging Technologies, 2009. ICET 2009. International Conference on*, pages 241–246. IEEE, 2009.

[Han02] Alan Hanjalic. Shot-boundary detection: unraveled and resolved? *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(2):90–105, 2002.

[Har62] Zellig S. Harris. *String analysis of sentence structure*. The Hague: Mouton, 1962.

[Har13] Olaf Hartig. SQUIN: a traversal based query execution system for the web of linked data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 1081–1084, 2013.

[Haw13] Sandro Hawke. SPARQL Query Results XML Format (Second Edition), 2013.

[HCL04] Yu-Hsuan Ho, Wei-Ren Chen, and Chia-Wen Lin. A rate-constrained key-frame extraction scheme for channel-aware video streaming. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 1, pages 613–616. IEEE, 2004.

[He05] Xiaofei He. *Locality Preserving Projections*. PhD thesis, University of Chicago, 2005.

[Hen01] Robbert Günter Henrich Andreas. POQLˆMM: A Query Language for Structured Multimedia Documents. *Proceedings 1st International Workshop on Multimedia Data and Document Engineering (MDDE'01)*, pages 22–229, 2001.

[HHB03] J. Huang, B. Heisele, and V. Blanz. Component-based face recognition with 3d morphable models. In *International Conference on Audio- and Video-Based Person Authentication*, 2003.

[Hin07] G.E. Hinton. Learning multiple layers of representation. *Trends in Cognitive Sciences*, 11:428–434, 2007.

[Hin09] G. E. Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[HKH99] Thorsten Hermes, Christoph Klauck, and Otthein Herzog. Knowledge-based image retrieval. In B. Jähne, H. Haussecker, and P. Geissler, editors, *Handbook of Computer Vision and Application*, volume 3, chapter 25, pages 515–530. Academic Press, 1999.

[HKKZ95] Thorsten Hermes, Christoph Klauck, Jutta Kreyß, and Jinyou Zhang. Content-based image retrieval. In Dennis Bockus, Karen Bennet, W. Morven Gentleman, J. Howard Johnson, and Evelyn Kidd, editors, *Proc. 1995 Conference of the Centre for Advanced Studies on Collaborative Research (CASCON 1995)*. IBM Press, 1995.

[HKLR81] L.D. Harmon, M.K. Khan, R. Lasch, and P.F. Raming. Machine identification of human faces. *Pattern Recognition*, 13:97–110, 1981.

[HKM+97] Jing Huang, S. Ravi Kumar, Mandar Mitra, Zhu Wei-Jing, and Ramin Zabih. Image indexing using color correlograms. In *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 762–768, Los Alamitos (Calif.) and and Brussels and and Washington [etc.], 1997. IEEE Computer Society.

[HL01] E. Hjelmas and B.K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83:236–274, 2001.

[HLAB13] Sebastian Hellmann, Jens Lehman, Sören Auer, and Martin Brümmer. Integrating NLP using linked data. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof Janowicz, editors, *Proc. 12th International Semantic Web Conference (ISWC 2013), Part II*, volume 8219 of *Lecture Notes in Computer Science*, pages 98–113. Springer, 2013.

[HOT06] G. E. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[HRBLM07] 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.

[HRR78] S.C. Harmon, L.D. Kuo, P.F. Raming, and U. Raudkivi. Identification of human face profiles by computers. *Pattern Recognition*, 10:301–312, 1978.

[HS81] Berthold K.P Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

[HS99] H. Hermansky and S. Sharma. Temporal patterns (TRAPS) in ASR of noisy speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 289–292, Phoenix, AZ, USA, 1999.

[HS13a] Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language, 2013.

[HS13b] Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language. W3C Recommendation 21 March 2013, 2013. `http://www.w3.org/TR/sparql11-query/`.

[HSSVdS11] B. Haslhofer, R. Simon, R. Sanderson, and H. Van de Sompel. The open annotation collaboration (OAC) model. In *Proc. 2011 Workshop on Multimedia on the Web (MMWeb 2011)*. IEEE Computer Society, 2011.

[HTG09] Quan Huynh-Thu and Mohammed Ghanbari. No-reference temporal quality metric for video impaired by frame freezing artefacts. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2221–2224. IEEE, 2009.

[HV13] Frank B. ter Haar and Remco Veltkamp. *3D Morphable Models for Face Surface Analysis and Recognition*, pages 119–147. John Wiley & Sons SingaporePte Ltd, 2013.

[HXL+11] Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank. A survey on visual content-based video indexing and retrieval. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 41(6):797–819, 2011.

[HYH+05] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):328–40, March 2005.

[HZ09] Feng Han and Song Chun Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):59–73, 2009.

[IK99] L. Itti and C. Koch. Target detection using saliency-based attention. In *Workshop on Search and Target Acquisition (NATO Unclassified)*, pages 3.1–3.10, Utrecht, The Netherlands, 1999.

[ISO99] ISO/IEC. Information technology – Database languages – SQL Multimedia and Application Packages – Part 3: Spatial. ISO 13249-3:1999, International Organization for Standardization, Geneve, Switzerland, 1999.

[ISO00a] ISO/IEC. Information technology – Database languages – SQL multimedia and application packages – Part 1: Framework. ISO 13249-1:2000, International Organization for Standardization, Geneve, Switzerland, 2000.

[ISO00b] ISO/IEC. Information technology – Database languages – SQL multimedia and application packages – Part 2: Full-Text. ISO 13249-2:2000, International Organization for Standardization, Geneve, Switzerland, 2000.

[ISO01] ISO/IEC. Information technology – Database languages – SQL multimedia and application packages – Part 5: Still Image. ISO 13249-5:2001, International Organization for Standardization, Geneve, Switzerland, 2001.

[ISO06] ISO/IEC. Information technology – Database languages – SQL multimedia and application packages – Part 6: Data mining. ISO 13249-6:2006, International Organization for Standardization, Geneve, Switzerland, 2006.

[ISO11] ISO/IEC. Information technology – Database languages – SQL – Part 11: Information and Definition Schemas (SQL/Schemata). ISO 9075-11:2011, International Organization for Standardization, Geneve, Switzerland, 2011.

[JA09] Rabia Jafri and Hamid R Arabnia. A Survey of Face Recognition Techniques. *Journal of Information Processing Systems*, 5(2):41–68, 2009.

[JAB+12] Bevan Jones, Jacob Andreas, Daniel Bauer, Karl Moritz Hermann, and Kevin Knight. Semantics-based machine translation with hyperedge replacement grammars. In *Proc. 24th International Conference on Computational Linguistics (COLING 2012): Technical Papers*, pages 1359–1376. The Association for Computer Linguistics, 2012.

[JER09] Cyril Joder, Slim Essid, and Gael Richard. Temporal integration for audio classification with application to musical instrument classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(1):174–186, 2009.

[JLMT07] Xin Jin, Ying Li, Teresa Mah, and Jie Tong. Sensitive webpage classification for content advertising. In *Proc. 1st International Workshop on Data Mining and Audience Intelligence for Advertising (ADKDD'07)*, pages 28–33. ACM, 2007.

[JM00] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 1st edition, 2000.

[JMP06] Audun Jøsang, Stephen Marsh, and Simon Pope. Exploring different types of trust propagation. In *Trust management*, pages 179–192. Springer, 2006.

[JOMM02] Roman Jarina, Noel E. O'Connor, Seán Marlow, and Noel Murphy. Rhythm detection for speech-music discrimination in mpeg compressed domain. In *Proceedings of the 14th IEEE International Conference on Digital Signal Processing*, 2002.

[Jon07] Yosi Mass Jonathan Mamou. A Query Language for Multimedia Content. In *Proceeding of the Multimedia Infomation Retreival workshop held in conjunction with the 30th Annual International ACM SIGIR Conceference*, 2007.

[KAC+02] Gregory Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. RQL: a declarative query language for RDF. In *Proc Intl World Wide Web Conf WWW*, pages 592–603, 2002.

[Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[Kan73] T. Kanade. *Computer Recognition of Human Faces*. Birkhauser, Basel, Switzerland, and Stuttgart, Germany, 1973.

[KB76] G.J. Kaufman and K.J. Breeding. Automatic recognition of human faces from profile silhouettes. *IEEE Transactions On Systems Man And Cybernetics (SMC)*, 6:113–121, 1976.

[KB13] Hjalmar S. Kühl and Tilo Burghardt. Animal Biometrics: Quantifying and Detecting Phenotypic Appearance. *Trends in Ecology & Evolution*, 28(7):432–441, March 2013.

[KC01] Irena Koprinska and Sergio Carrato. Temporal video segmentation: A survey. *Signal processing: Image communication*, 16(5):477–500, 2001.

[KE06] C. Küblbeck and A. Ernst. Face detection and tracking in video sequences using the modified census transformation. *Image and Vision Computing*, 24(6):564–572, 2006.

[Kel70] M. D. Kelly. Visual identification of people by computer. Technical report, Stanford AI Project, Stanford, CA, 1970.

[KH08] Y Kawayokeita and Yuukou Horita. Nr objective continuous video quality assessment model based on frame quality measure. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 385–388. IEEE, 2008.

[KHC07] Jacek P. Kukluk, Lawrence B. Holder, and Diane J. Cook. Inference of node replacement graph grammars. *Intelligent Data Analysis*, 11(4):377–400, 2007.

[KHC08] Jacek P. Kukluk, Lawrence B. Holder, and Diane J. Cook. Inference of edge replacement graph grammars. *International Journal on Artificial Intelligence Tools*, 17(3):539–554, 2008.

[KHWB05] Zia Khan, Rebecca A. Herman, Kim Wallen, and Tucker Balch. An Outdoor 3-D Visual Tracking System for the Study of Spatial Navigation and Memory in Rhesus Monkeys. *Behavior Research Methods*, 37(3):453–463, 2005.

[KIKY09] Tatsuo Kozakaya, Satoshi Ito, Susumu Kubota, and Osamu Yamaguchi. Cat Face Detection with Two Heterogeneous Features. In *International Conference on Image Processing (ICIP)*, pages 1213–1216, Cairo, Egypt, November 2009. Ieee.

[Kla94] Christoph Klauck. *Eine Graphgrammatik zur Repräsentation und Erkennung von Features in CAD/CAM*. PhD thesis, University of Kaiserslautern, 1994.

[KM03] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proc. 41st Annual Meeting of the Association for Computational Linguistics (ACL'03) – Volume 1*, pages 423–430. The Association for Computational Linguistics, 2003.

[KM12] Krishna Kulkarni and Jan-Eike Michels. Temporal features in SQL:2011. *ACM SIGMOD Record*, 41(3):34, October 2012.

[KMF11] Mayumi Kouda, Masakazu Morimoto, and Kensaku Fujii. A Face Identification Method of Non-Native Animals for Intelligent Trap. In *Conference on Machine Vision Applications (MVA)*, number 4, pages 426–429, Nara, Japan, 2011.

[KMN09] Sandra Kübler, Ryan McDonald, and Joakim Nivre. *Dependency Parsing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool, 2009.

[KMW+11] Gerald Kastberger, Michael Maurer, Frank Weihmann, Matthias Ruether, Thomas Hoetzl, Ilse Kranner, and Horst Bischof. Stereoscopic Motion Analysis in Densely Packed Clusters: 3D Analysis of the Shimmering Behaviour in Giant Honey Bees. *Frontiers in Zoology*, 8(3):2–18, January 2011.

[Knu68] Donald Knuth. Semantics of context-free languages. *Mathematical Systems Theory*, 2(2):127–145, 1968.

[Kos86] Bart Kosko. Fuzzy cognitive maps. *International Journal of man-machine studies*, 24(1):65–75, 1986.

[KOS11] D. R. Karger, S. Oh, and D. Shah. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. *ArXiv e-prints*, October 2011.

[KP12] Isaak Kavasidis and Simone Palazzo. Quantitative Performance Analysis of Object Detection Algorithms on Underwater Video Footage. In *ACM International Workshop on Multimedia Analysis for Ecological Data (MAED)*, pages 57 – 60, Nara, Japan, 2012.

[KS04] Yan Ke and Rahul Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 506–513, 2004.

[KS12] Yogan Jaya Kumar and Naomie Salim. Automatic multi document summarization approaches. *Journal of Computer Science*, 8(1):133–140, 2012.

[KSWP05] Franc Kozamernik, Paola Sunna, Emmanuel Wyckens, and Dag Inge Pettersen. Subjective quality of internet video codecs phase ii evaluations using samviq. *EBU technical Review*, 301, 2005.

[KV05] Changick Kim and Bhaskaran Vasudev. Spatiotemporal sequence matching for efficient video copy detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):127–132, 2005.

[KWH12] Pooya Khorrami, Jiangping Wang, and Thomas Huang. Multiple Animal Species Detection Using Robust Principal Component Analysis and Large Displacement Optical Flow. In *Workshop on Visual Observation and Analysis of Animal and Insect Behavior (VAIB)*, Tsukuba, Japan, 2012.

[KYS09] Helen Kwong and Neil Yorke-Smith. Detection of imperative and declarative question-answer pairs in email conversations. In Craig Boutilier, editor, *Proc. 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 1519–1524. Morgan Kaufmann, 2009.

[KZZ06] Jun Kong, Kang Zhang, and Xiaoqin Zeng. Spatial graph grammars for graphical user interfaces. *ACM Transactions on Computer-Human Interaction*, 13(2):268–307, 2006.

[LB06] Mohsen Lesani and Saeed Bagheri. Applying and inferring fuzzy trust in semantic web social networks. In *Canadian Semantic Web*, pages 23–43. Springer, 2006.

[LCC95] A. Lanitis, Taylor. C.J., and T.F. Cootes. Automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13:393–401, 1995.

[LCH01] Peiya Liu, Amit Chakraborty, and Liang H Hsu. A Logic Approach for MPEG-7 XML Document Queries. In *Proceedings of Extreme Markup Languages®*, 2001.

[LCLS10] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[LCS11] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 IEEE International Conference on Computer Vision*, pages 2548–2555, 2011.

[LDZ+07] Enrique Larios, Hongli Deng, Wei Zhang, Matt Sarpola, Jenny Yuen, Robert Paasch, Andrew Moldenke, David Lytle, Salvador Ruiz Correa, Eric Mortensen, Linda Shapiro, Tom Dietterich, and Feature Vector Generation. Automated Insect Identification through Concatenated Histograms of Local Appearance Features. In *IEEE Workshop on Applications of Computer Vision (WACV)*, Austin, Texas, USA, 2007.

[LG13] Markus Lanthaler and Christian Guetl. Hydra: A vocabulary for hypermedia-driven web APIs. In *Proc. WWW 2013 Workshop on Linked Data on the Web (LDOW 2013)*, volume 996 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.

[LGLW09] Liang Lin, Haifeng Gong, Li Li, and Liang Wang. Semantic event representation and recognition using syntactic attribute graph grammar. *Pattern Recognition Letters*, 30(2):180–186, 2009.

[Lie01] Rainer Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics*, 1(03):469–486, 2001.

[LJCM13] JP Lopez, D Jimenez, A Cerezo, and JM Menendez. No-reference algorithms for video quality assessment based on artifact evaluation in mpeg-2 and h. 264 encoding standards. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 1336–1339. IEEE, 2013.

[LKH10] Hantao Liu, Nick Klomp, and Ingrid Heynderickx. A no-reference metric for perceived ringing artifacts in images. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(4):529–539, 2010.

[LKL97] S.H. Lin, S.Y. Kung, and L.J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Transactions on Neural Networks*, 8:114–132, 1997.

[LL99a] Z. Liposcak and S. Loncaric. Face recognition from profiles using morphological signature transform. In *21st International Conference Information Technology Interfaces*, Pula, Croatia, 1999.

[LL99b] Z. Liposcak and S. Loncaric. A scale-space approach to face recognition from profiles. In *8th International Conference on Computer Analysis of Images and Patterns*, 1999.

[LLWH12] Yanqiang Lei, Weiqi Luo, Yuangen Wang, and Jiwu Huang. Video sequence matching based on the invariance of color correlation. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(9):1332–1343, 2012.

[LLX13] Hong Liu, Hong Lu, and Xiangyang Xue. A segmentation and graph-based video sequence matching method for video copy detection. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1706–1718, 2013.

[LMA99] Nailja Luth, Andrea Miene, and Peter Alshuth. Syntactical and semantical description of video sequences. In Robert Meersman, Zahir Tari, and Scott M. Stevens, editors, *Proc. IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics – Semantic Issues in Multimedia Systems (DS-8)*, volume 138 of *IFIP Conference Proceedings*, pages 65–84. Kluwer, 1999.

[LMMZ+10] David A. Lytle, Gonzalo Marinez-Munoz, Wei Zhang, Natalia Larios, Linda Shapiro, Robert Paasch, Andrew Moldenke, Eric N. Mortensen, Sinisa Todorivic, and Thomas G. Dietterich. Automated Processing and Identification of Benthic Invertebrate Samples. *Journal of the North American Benthological Society*, 29(3):867–874, 2010.

[LOSO97] John Z. Li, M Tamer Özsu, Duane Szafron, and Vincent Oria. MOQL: A Multimedia Object Query Language. In *in Proceedings of the 3RD international workshop on Multimedia Information System*, 1997.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LSHN05] Lucian Vlad Lita, Andrew Hazen Schlaikjer, Weichang Hong, and Eric Nyberg. Qualitative dimensions in question answering: Extending the definitional QA task. In *Proc. 20th National Conference on Artificial Intelligence (AAAI'05) – Volume 4*, pages 1616–1617. The AAAI Press, 2005.

[LSY03] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, Jan 2003.

[LVB+93] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.

[LWO+12] Yunjia Li, Mike Wald, Tope Omitola, Nigel Shadbolt, and Gary Wills. Synote: weaving media fragments and linked data. 2012.

[LYK09] Sunil Lee, Chang D. Yoo, and Ton Kalker. Robust video fingerprinting based on symmetric pairwise boosting. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(9):1379–1388, 2009.

[LZ01] Rainer Lienhart and Andre Zaccarin. A system for reliable dissolve detection in videos. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 406–409. IEEE, 2001.

[LZ09] Tsau Young Lin and Shangxuan Zhang. An automata based authorship identification system. In Sanjay Chawla, Takashi Washio, Shin-ichi Minato, Shusaku Tsumoto, Takashi Onoda, Seiji Yamada, and Akihiro Inokuchi, editors, *Proc. PAKDD 2009 Workshop New Frontiers in Applied Data Mining*, volume 5433 of *Lecture Notes in Artificial Intelligence*, pages 134–142. Springer, 2009.

[Mac07] N. MacLeod. Automated taxon identification in systematics: Theory, approaches and applications. *Systematics Association Special Volume*, 74, 2007.

[Mar04a] M. Marchiori. Towards a People's Web: Metalog. *IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, 2004.

[Mar04b] Jose Maria Martinez. Mpeg-7 overview (version 10), iso. Technical report, IEC JTC1/SC29/WG11, 2004.

[May12a] Mark T. Maybury, editor. *Multimedia Information Extraction: Advances in Video, Audio, and Imagery Analysis for Search, Data Mining, Surveillance, and Authoring*. Wiley, 2012.

[May12b] Mark T. Maybury. Multimedia information extraction: History and state of the art. In Mark T. Maybury, editor, *Multimedia Information Extraction: Advances in Video, Audio, and Imagery Analysis for Search, Data Mining, Surveillance, and Authoring*, chapter 2, pages 13–40. Wiley, 2012.

[MBH+04] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. OWL-S: Semantic markup for web services. W3C Member Submission. 22 November, 2004. `http://www.w3.org/Submission/OWL-S/`.

[MCF+11] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, et al. The open provenance model core specification (v1. 1). *Future Generation Computer Systems*, 27(6):743–756, 2011.

[ME01] Jim Melton and Andrew Eisenberg. SQL Multimedia and Application Packages (SQL/MM). *SIGMOD Rec.*, 30(4):97–102, December 2001.

[ME05] Michael Mandel and Daniel P. W. Ellis. Song-level features and support vector machines for music classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 2005.

[MHB+10] Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Computer Vision – ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 183–196, Berlin and Heidelberg, 2010. Springer Berlin Heidelberg.

[MKP02] J.M. Martinez, R. Koenen, and F. Pereira. MPEG-7: The Generic Multimedia Content Description Standard, part 1. *IEEE Multimedia*, 9, 2002.

[MKP03] H. Mak, I. Koprinska, and J. Poon. Intimate: a web-based movie recommender using text categorization. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, pages 602–605, Oct 2003.

[MLB+13] Iacopo Masi, Giuseppe Lisanti, Andrew D. Bagdanov, Pietro Pala, and Alberto Del Bimbo. Using 3d models to recognize 2d faces in the wild. In *CVPR International Workshop on Socially Intelligent Surveillance and Monitoring (SISM)*, Portland, OR, USA, 2013.

[MN07] Wim Martens and Joachim Niehren. On the minimization of XML Schemas and tree automata for unranked trees. *Journal of Computer and System Sciences*, 73(4):550–583, 2007.

[MO11] Anna Margolis and Mari Ostendorf. Question detection in spoken conversations using textual conversations. In *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, – Short Papers*, pages 118–124. The Association for Computer Linguistics, 2011.

[MOVY01] B. S. Manjunath, Jens-Rainer. Ohm, V. Vinod Vasudevan, and Akio Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

[MR09] Christopher D. Manning and Prabhakar Raghavan. *An Introduction to Information Retrieval*. Cambridge University Press, 2009.

[MS05] Krystian Mikolajczyk and Cordelia Schmid. Performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[MSM93] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[MSV12] Bernardo Miranda, Joaquin Salas, and Pablo Vera. Bumblebees Detection and Tracking. In *Workshop on Visual Observation and Analysis of Animal and Insect Behavior (VAIB)*, 2012.

[NK13] Taeyoung Na and Munchurl Kim. A novel no-reference psnr estimation method with regard to de-blocking filtering effect in h. 264/avc bitstreams. 2013.

[NM11] Ani Nenkova and Kathleen McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103–233, 2011.

[NMF+98] Milind R Naphade, Rajiv Mehrotra, A Müfit Ferman, Jim Warnick, Thomas S Huang, and A Murat Tekalp. A high-performance shot boundary detection algorithm using multiple cues. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 884–887. IEEE, 1998.

[NS07] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[OCK+03] Jens-Rainer Ohm, Leszek Cieplinski, Heon Jun Kim, Santhana Krishnamachari, BS Manjunath, Dean S Messing, and Akio Yamada. The mpeg-7 color descriptors. 2003.

[OGGW00] M.A. O'Neill, I.D. Gauld, K.J. Gaston, and P. Weeks. Daisy: An automated invertebrate identification system using holistic vision techniques. In *Inaugural Meeting BioNET-INTERNATIONAL Group for Computer-Aided Taxonomy (BIGCAT)*, pages 13–22, 2000.

[Ohm01] Jens-Rainer Ohm. The mpeg-7 visual description framework — concepts, accuracy, and applications. In *Computer Analysis of Images and Patterns*, volume 2124 of *Lecture Notes in Computer Science*, pages 2–10, Berlin and Heidelberg, 2001. Springer Berlin Heidelberg.

[O'N07] O'Neill. *DAISY: A Practical Computer-Based Tool for Semi-Automated Species Identification, Automated Taxon Identification in Systematics, Theory Approaches and Applications*, chapter 7, pages 101–114. N. MacLeod (Ed.) CRC Press, 2007.

[O'N10] MA O'Neill. DAISY: A Practical Tool for Semi-Automated Species Identification. Technical report, Department of Biology, Newcastle, UK, Newcastle, UK, 2010.

[OOX+99] V. Oria, M.T. Ozsu, Bing Xu Bing Xu, I. Cheng, and P.J. Iglinski. VisualMOQL: the DISIMA visual query language. *Proceedings IEEE International Conference on Multimedia Computing and Systems*, 1, 1999.

[Ope99] Open GIS Consortium, Inc. OpenGIS simple features specification for SQL. *OpenGIS Project Document 99*, 49:49–99, 1999.

[OPH96] T. Ojala, M. Pietikäinen, and D Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[OTO05] Marek R. Ogiela, Ryszard Tadeusiewicz, and Lidia Ogiela. Intelligent semantic information retrieval in medical pattern cognitive analysis. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganá, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *Proc. International Conference on Computational Science and Its Applications (ICCSA 2005), Part IV*, volume 3483 of *Lecture Notes in Computer Science*, pages 852–857. Springer, 2005.

[PA96] P. Penev and J. Atick. Local feature analysis: A general statistical theory for objecct representation. *Network: Computation in Neural Systems*, 7(3):477–500, 1996.

[PAG09] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of SPARQL. *ACM Transactions on Database Systems*, 34(3):1–45, August 2009.

[PBRT99] Jan Puzicha, Joachim M. Buhmann, Yossi Rubner, and Carlo Tomasi. Empirical evaluation of dissimilarity measures for color and texture. In *1999 IEEE International Conference on Computer Vision*, pages 1165–1172 vol.2, Los Alamitos and CA, 1999. IEEE Computer Society.

[PJS11] Matthew Perry, Prateek Jain, and AmitP. Sheth. Sparql-st: Extending sparql to support spatiotemporal queries. In Naveen Ashish and Amit P. Sheth, editors, *Geospatial Semantics and the Semantic Web*, volume 12 of *Semantic Web and Beyond*, pages 61–86. Springer US, 2011.

[PJW02] Mira Park, Jesse S. Jin, and Laurence S. Wilson. Fast content-based image retrieval using quasi-gabor filter and reduction of image feature dimension. In *Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 178–182, 2002.

[PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008.

[PMS94] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994.

[POM06] F. Porikli, Tuzel O., and P. Meer. Covariance tracking using model update based on lie algebra. In *Computer Vision and Pattern Recognition (CVPR)*, pages 728–735, New York, NY, USA, 2006.

[Por05] F. Porikli. Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images. In *IEEE Workshop on Motion and Video Computing*, pages 20–27, Breckenridge, CO, USA, 2005.

[Por06] F. Porikli. Achieving real-time object detection and tracking under extreme conditions. *Journal on Real-Time Image Processing*, 1(1):33–40, 2006.

[Pru04] Eric Prud'hommeaux. Algae RDF Query Language, 2004.

[PS13] Denis Parra and Shaghayegh Sahebi. Recommender systems: Sources of knowledge and evaluation metrics. In *Advanced Techniques in Web Intelligence-2*, pages 149–175. Springer, 2013.

[PT05] C. Panagiotakis and G. Tziritas. A speech/music discriminator based on rms and zero-crossings. *IEEE Transactions on Multimedia*, 7(1):155–166, 2005.

[PVJZ11] Omkar M Parkhi, Andrea Vedaldi, C V Jawahar, and Andrew Zisserman. The Truth About Cats and Dogs. In *International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011.

[RBG+13] M. J. Raddick, G. Bracey, P. L. Gay, C. J. Lintott, C. Cardamone, P. Murray, K. Schawinski, A. S. Szalay, and J. Vandenberg. Galaxy Zoo: Motivations of Citizen Scientists. *ArXiv e-prints*, March 2013.

[RBK98] H.A. Rowley, S. Baluja, and T. Kanade. Neural network- based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision ECCV 2006*, volume 3951, pages 430–443, 2006.

[RDHP07] Kimberly N. Russell, Martin T. Do, Jeremy C. Huff, and Norman I. Platnick. Introducing SPIDA-Web: Wavelets, Neural Networks and Internet Accessibility in an Image-Based Automated Identification System. In N. (Ed) MacLeod, editor, *Automated Taxon Identification in Systemtatics: Theory, Approaches, and Applications*, chapter Chapter 9, pages 131–152. 2007.

[RF03] Deva Ramanan and D.A. Forsyth. Using Temporal Coherence to Build Models of Animals. In *International Conference on Computer Vision (ICCV)*, pages 1–8, Nice, France, 2003.

[RFB05] D. Ramanan, D. A. Forsyth, and K. Barnard. Detecting, Localizing and Recovering Kinematics of Textured Animals. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 635–642, San Diego, California, USA, 2005. Ieee.

[RFB06] Deva Ramanan, David A. Forsyth, and Kobus Barnard. Building Models of Animals from Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(8):1319–34, August 2006.

[RIS+94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM.

[RKL+05] Dumitru Roman, Uwe Keller, Holger Lausen, Jos de Bruijn, Rubén Lara, Michael Stollberg, Axel Polleres, Cristina Feier, Christoph Bussler, and Dieter Fensel. Web service modeling ontology. *Applied Ontology*, 1(1):77–106, 2005.

[RKM10] Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. Authorship attribution using probabilistic context-free grammars. In *Proc. 48th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, – Short Papers*, pages 38–42. The Association for Computational Linguistics, 2010.

[Rod10] MTA Rodrigues. Automatic Fish Species Classification Based on Robust Feature Extraction Techniques and Artificial Immune Systems. In *International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, number i, pages 1518–1525, Liverpool, UK, 2010.

[RRB12] S. Roy, S. Roy, and S.K. Bandyopadhyay. A tutorial review on face detection. *International Journal of Engineering Research & Technology (IJERT)*, 1(8), 2012.

[RRKB11] E. Rublee, V. Rabaud, K. Konolige, and G.R. Bradski. Orb: An efficient alternative to sift or surf. In *International Conference on Computer Vision (ICCV)*, pages 2564–2571, Barcelona, Spain, 2011.

[RRS11] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[RS03] Zeeshan Rasheed and Mubarak Shah. Scene detection in hollywood movies and tv shows. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–343. IEEE, 2003.

[RT12a] Giuseppe Rizzo and R Troncy. NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. *EACL 2012*, pages 73–76, 2012.

[RT12b] Giuseppe Rizzo and Raphaël Troncy. NERD: A framework for unifying named entity recognition and disambiguation web extraction tools (poster). In *13th Conference of the European Chapter of the Association for Computational Linguistics (EACL'12)*. The Association for Computational Linguistics, 2012.

[RTHB12] G. Rizzo, R. Troncy, S. Hellmann, and M. Bruemmer. NERD meets NIF: Lifting NLP extraction results to the linked data cloud. In Christian Bizer, Tom Heath, Tim Berners-Lee, and Michael Hausenblas, editors, *Proc. WWW 2012 Workshop on Linked Data on the Web (LDOW 2012)*, volume 937 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[RW10] Abdul Rehman and Zhou Wang. Reduced-reference ssim estimation. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 289–292. IEEE, 2010.

[RWP05] Ellen Riloff, Janyce Wiebe, and William Phillips. Exploiting subjectivity classification to improve information extraction. In *Proc. 20th National Conference on Artificial Intelligence (AAAI'05) – Volume 3*, pages 1106–1111. The AAAI Press, 2005.

[SA07] Evaggelos Spyrou and Yannis Avrithis. Keyframe extraction using local visual semantics in the form of a region thesaurus. In *Semantic Media Adaptation and Personalization, Second International Workshop on*, pages 98–103. IEEE, 2007.

[Sak90] Yasubumi Sakakibara. Learning context-free grammars from structural data in polynomial time. *Theoretical Computer Science*, 76:223–242, 1990.

[SB91] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[SB12] Michele A Saad and Alan C Bovik. Blind quality assessment of videos using a model of natural scene statistics and motion coherency. In *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, pages 332–336. IEEE, 2012.

[SB13] Roz Sandwell and Tilo Burghardt. Chimpanzee Face Detection: An Automated System for Images Captured From Natural Environments. In *International Conference on Behaviour, Physiology and Genetics of Wildlife*, Berlin, Germany, 2013.

[SBB+13] Florian Stegmaier, Werner Bailer, Tobias Bürger, Mari Carmen Suárez-Figueroa, Erik Mannens, Jean-Pierre Evain, Martin Höffernig, Pierre Champin, Mario Döller, and Harald Kosch. Unified access to media metadata on the Web. *IEEE MultiMedia*, 20(2):22–29, 2013.

[SBK+12] Sebastian Schaffert, Christoph Bauer, Thomas Kurz, Fabian Dorschel, Dietmar Glachs, and Manuel Fernandez. The Linked Media Framework: Integrating and Interlinking Enterprise Media Content and Data. *Proceedings of the 8th International Conference on Semantic Systems - I-SEMANTICS '12*, 2012.

[SCBNF08] C. Spampinato, Y.H. Chen-Burger, G. Nadarajan, and R.B. Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. In *VISAPP*, pages 514–519, 2008.

[Sch01a] C. Schmid. Constructing Models for Content-Based Image Retrieval. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 39–45, Kauai, Hawaii, USA, 2001.

[Sch01b] S. Schröder. *Automatisierte Identifikation von Bienenarten (Apidae, Hymenoptera) anhand ihres Flügelgeäders durch Methoden der digitalen Bildverarbeitung und der statistischen Klassifikation*. Phd thesis, University of Bonn, 2001.

[Sch04] Sebastian Schaffert. *Xcerpt: A Rule-Based Query and Transformation Language for the Web*. PhD thesis, University of Munich, 2004.

[SDB+04] Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. The ICSI meeting recorder dialog act (MRDA) corpus. In *In Proc. 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100. The Association for Computational Linguistics, 2004.

[Sea04] Andy Seaborne. RDQL - A Query Language for RDF (Member Submission), 2004.

[SEN98] J. Steffens, E. Elagin, and H. Neven. Personspotter - fast and robust system for human detection, tracking and recognition. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, Washington, DC, USA, 1998.

[SES07] Johannes Stallkamp, Hazim K. Ekenel, and Rainer Stiefelhagen. Video-based Face Recognition on Real-World Data. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. Ieee, October 2007.

[SFHS07] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[SG10] Fangxia Shi and Xiaojun Guo. Keyframe extraction based on kmeas results to adjacent dc images similarity. In *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, volume 1, pages V1–611. IEEE, 2010.

[SGD+10] Concetto Spampinato, Daniela Giordano, Roberto Di Salvo, Yun-Heh Chen-Burger, Robert B. Fisher, and Gayathri Nadarajan. Automatic Fish Classification for Underwater Species Behavior Understanding. In *ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams (ARTEMIS)*, pages 45–50, Firenze, Italy, 2010. ACM Press.

[SKKR01] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[SKR99] J. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, pages 158–166, New York, NY, USA, 1999. ACM.

[Sla94] Malcolm Slaney. Auditory toolbox, 1994.

[SM97] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[SM07] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Minneapolis, MN, USA, 2007.

[SM12] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.

[Sma09] Kevin Small. *Interactive Learning Protocols for Natural Language Applications*. PhD thesis, University of Illinois, 2009.

[SOD10] Alan F Smeaton, Paul Over, and Aiden R Doherty. Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding*, 114(4):411–418, 2010.

[SOK06] Alan F. Smeaton, Paul Over, and Wessel 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. ACM Press.

[SPB+12] Concetto Spampinato, Simone Palazzo, Bastian Boom, Jacco van Ossenbruggen, Isaak Kavasidis, Roberto Di Salvo, Fang-Pang Lin, Daniela Giordano, Lynda Hardman, and Robert B. Fisher. Understanding Fish Behavior During Typhoon Events in Real-Life Underwater Environments. *Multimedia Tools and Applications*, 68(1), 2012.

[Spe97] Ellen Spertus. Smokey: Automatic recognition of hostile messages. In Benjamin Kuipers and Bonnie L. Webber, editors, *Proc. 9th Conference on Innovative Applications of Artificial Intelligence (IAAI'97)*, pages 1058–1065. The AAAI Press, 1997.

[SPT06] Wolf Siberski, JeffZ. Pan, and Uwe Thaden. Querying the semantic web with preferences. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and LoraM. Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 612–624. Springer Berlin Heidelberg, 2006.

[SQX+06] Dezhen Song, Ni Qin, Yiliang Xu, Chang Young Kim, David Luneau, and Ken Goldberg. System and Algorithms for an Autonomous Observatory Assisting the Search for the Ivory-Billed Woodpecker. In *International Conference on Automation Science and Engineering (CASE)*, pages 200–205, Shanghai, China, 2006.

[SR12] Robert Sumner and Laura Ross. Extension of the Viola-Jones Detector – Application to Cat Faces. Technical report, Boston University, Boston, 2012.

[SRPS12] E. Simpson, S. Roberts, I. Psorakis, and A. Smith. Dynamic Bayesian Combination of Multiple Imperfect Classifiers. *ArXiv e-prints*, June 2012.

[SS97] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/-music discriminator. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 1331–1334, 1997.

[SS99] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[SS10] Carsten Saathoff and Ansgar Scherp. Unlocking the semantics of multimedia presentations in the web with the multimedia metadata ontology. In *Proceedings of the 19th international conference on World wide web*, pages 831–840. ACM, 2010.

[SSBC10] Kalpana Seshadrinathan, Rajiv Soundararajan, Alan Conrad Bovik, and Lawrence K Cormack. Study of subjective and objective quality assessment of video. *Image Processing, IEEE transactions on*, 19(6):1427–1441, 2010.

[SSH05] Ingo Schmitt, Nadine Schulz, and Thomas Herstel. WS-QBE: A QBE-Like Query Language for Complex Multimedia Queries. *11th International Multimedia Modelling Conference*, 2005.

[SSH12] Henry Stahl, Kristina Schädler, and Eberhard Hartung. Capturing 2D and 3D Biometric Data of Farm Animals under Real-Life Conditions. In *International Workshop on Computer Image Analysis in Agriculture*, number 17, Valencia, Spain, 2012.

[SSS02] Deepak Sirdeshmukh, Jagdip Singh, and Barry Sabol. Consumer trust, value, and loyalty in relational exchanges. *Journal of marketing*, 66(1):15–37, 2002.

[ST94] J. Shi and C. Tomasi. Good features to track. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, Seattle, Washington, USA, 1994.

[Ste00] Volker Steinhage. Automated Identification of Bee Species in Biodiversity Information Systems. *Computer Science for Environmental Protection*, 1(339-344):1–6, 2000.

[STL04] Gary J Sullivan, Pankaj N Topiwala, and Ajay Luthra. The h. 264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions. In *Optical Science and Technology, the SPIE 49th Annual Meeting*, pages 454–474. International Society for Optics and Photonics, 2004.

[Sto03] Knut Stolze. SQL/MM Spatial - The Standard to Manage Spatial Data in a Relational Database System. In *BTW*, pages 247–264, 2003.

[Suk00] G. Sukthankar. Face recognition: a critical look at biologically-inspired approaches. Cmuri- tr-00-04, Carnegie Mellon University, Pittsburgh, PA, USA, 2000.

[SWJ+09] John L A Salle, Quentin Wheeler, Paul Jackway, Shaun Winterton, and David Lovell. Accelerating Taxonomic Discovery Through Automated Character Extraction. *Zootaxa*, 2217:43–55, 2009.

[SWS+00a] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[SWS+00b] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000.

[TDMP12] Raphaël Troncy, Davy Van Deursen, Erik Mannens, and Silvia Pfeiffer. Media Fragments URI 1.0 (basic). W3C recommendation, W3C, September 2012.

[TEBEH06] A. S. Tolba, A. H. El-Baz, and A.A. El-Harby. Face Recognition: A Literature Review. *International Journal of Information and Communication Engineering*, 2(2):88–103, 2006.

[Tho12] Herbert Thoma. A system for subjective evaluation of audio, video and audiovisual quality using mushra and samviq methods. In *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*, pages 31–32. IEEE, 2012.

[TLF10] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010.

[TLT+13] H Tan, Zhengguo Li, Y Tan, Susanto Rahardja, and Chuohuo Yeo. A perceptually relevant mse-based image quality metric. 2013.

[TMY78] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473, 1978.

[TNC10] Hung-Khoon Tan, Chong-Wah Ngo, and Tat-Seng Chua. Efficient mining of multiple partial near-duplicate alignments by temporal network. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(11):1486–1498, 2010.

[Tof08] Adam Tofilski. Using Geometric Morphometrics and Standard Morphometry to Discriminate Three Honeybee Subspecies. *Apidologie*, 39(5):558–563, 2008.

[Ton01] Richard Tong. An operational system for detecting and tracking opinions in on-line discussions. In *Working Notes of the SIGIR Workshop on Operational Text Classification*, pages 1–6, New Orleans, Louisianna, 2001.

[TP91a] Matthew Turk and Alex Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71 – 86, 1991.

[TP91b] Matthew A. Turk and Alex P. Pentland. Face Recognition Using Eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[TPJL05] WT Teacy, Jigar Patel, Nicholas R Jennings, and Michael Luck. Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 997–1004. ACM, 2005.

[Tra01] Markus Trauberg. *Ein Verfahren zur Qualitätsbewertung datenreduzierter Bildfolgen*. Shaker, 2001.

[TT10] Xiaoyang Tan and Bill Triggs. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, 2010.

[TV07] Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3(1):3, 2007.

[TVD03] Ba Tu Truong, Svetha Venkatesh, and Chitra Dorai. Scene extraction in motion pictures. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(1):5–15, 2003.

[TYRW14] Y. Taigman, M. Yang, M.A. Ranzato, and L. Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[TZ04] Wallapak Tavanapong and Junyu Zhou. Shot clustering techniques for story browsing. *Multimedia, IEEE Transactions on*, 6(4):517–527, 2004.

[UF11] Muhammad Uzair and Dalia Fayek. An efficient no-reference blockiness metric for intra-coded video frames. In *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, pages 1–5. IEEE, 2011.

[Uta12] Akos Utasi. Local Appearance Feature Based Classification of the Theraphosidae Family. In *Visual Observation and Analysis of Animal and Insect Behavior (VAIB)*, Tsukuba, Japan, 2012.

[VDCC11] Patricia Victor, Martine De Cock, and Chris Cornelis. Trust and recommendations. In *Recommender systems handbook*, pages 645–675. Springer, 2011.

[Vij13] V. Vijayakumari. Face Recognition Techniques: A Survey. *World Journal of Computer Application and Technology*, 1(2):41–50, 2013.

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, Kauai, Hawai, USA, 2001.

[Vla00] T Vlachos. Detection of blocking artifacts in compressed video. *Electronics Letters*, 36(13):1106–1108, 2000.

[VMM+07] Adriano Veloso, Wagner Meira, Tiago Macambira, Dorgival Guedes, and Hélio Almeida. Automatic moderation of comments in a large on-line journalistic environment. In Natalie S. Glance, Nicolas Nicolov, Eytan Adar, Matthew Hurst, Mark Liberman, and Franco Salvetti, editors, *Proc. 1st International Conference on Weblogs and Social Media (ICWSM'07)*, 2007.

[VZ03] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[W3C13] W3C. W3C Data Activity. `http://www.w3.org/2013/data/`, 2013.

[WB06] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Department of Computer Science, University of North Carolina, 2006.

[WBL02] Zhou Wang, Alan C Bovik, and Ligang Lu. Why is image quality assessment so difficult? In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–3313. IEEE, 2002.

[WBSS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[WC10] Kai Wang and Tat-Seng Chua. Exploiting salient patterns for question detection and question retrieval in community-based question answering. In Chu-Ren Huang and Dan Jurafsky, editors, *Proc. 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 1155–1163. Tsinghua University Press, 2010.

[WEK04] Dirk Walther, Duane R Edgington, and Christof Koch. Detection and Tracking of Objects in Underwater Video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 544–549, Washington, District Columbia, USA, 2004.

[WFKvdM97] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):775–779, 1997.

[WGY03] W. Q. Wang, W. Gao, and D. W. Ying. A fast and robust speech/music discrimination approach. In *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and Proceedings of the Fourth Pacific Rim Conference on Multimedia.*, volume 3, pages 1325–1329, 2003.

[Wil07] Gregory Todd Williams. Extensible SPARQL functions with embedded Javascript. volume 248 of *CEUR Workshop Proceedings ISSN 1613-0073*, June 2007.

[Wil13] Gregory Todd Williams. SPARQL 1.1 Service Description. W3C Recommendation 21 March 2013, 2013. `http://www.w3.org/TR/sparql11-service-description/`.

[WIY09] T. Watanabe, S. Ito, and K. Yokoi. Co-occurrence histograms of oriented gradients for pedestrian detection. *PSIVT, LNCS*, 5414:37–47, 2009.

[WMM+09] Jens Wawerla, Shelley Marshall, Greg Mori, Kristina Rothley, and Payam Sabzmeydani. BearCam: Automated Wildlife Monitoring at the Arctic Circle. *Machine Vision and Applications*, 20(5):303–317, April 2009.

[WSL+13] Michael J. Wilber, Walter J. Scheirer, Phil Leitne, BrianBoult, James Zott, Daniel Reinke, David Delaney, and Terrance Bolt. Animal Recognition in the Mojave Desert: Vision Tools for Field Biologists. In *Workshop on the Applications of Computer Vision (WACV)*, Clearwater Beach, Florida, USA, 2013.

[WV03] Yao Wang and Julita Vassileva. Trust and reputation model in peer-to-peer networks. In *Peer-to-Peer Computing, 2003.(P2P 2003). Proceedings. Third International Conference on*, pages 150–157. IEEE, 2003.

[WYG+09] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–27, February 2009.

[WYY+10] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3360–3367, San Francisco, CA, USA, 2010.

[XHS+10] Qing Xie, Zi Huang, Heng Tao Shen, Xiaofang Zhou, and Chaoyi Pang. Efficient and continuous near-duplicate video detection. In *2010 12th Asia Pacific Web Conference (APWEB)*, pages 260–266, 2010.

[YCCY13] Daode Yang, Sikan Chen, Yuanhui Chen, and Yuying Yan. Using Head Patch Pattern as a Reliable Biometric Character for Noninvasive Individual Recognition of an Endangered Pitviper Protobothrops Mangshanensis. *Asian Herpetological Research*, 4(2):134–139, 2013.

[YKA02] M.H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[YRH+10] Junyong You, Ulrich Reiter, Miska M Hannuksela, Moncef Gabbouj, and Andrew Perkis. Perceptual-based quality assessment for audio–visual services: A survey. *Signal Processing: Image Communication*, 25(7):482–501, 2010.

[YWK10] Gilbert Yammine, Eugen Wige, and André Kaup. A no-reference blocking artifacts visibility estimator in images. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2497–2500. IEEE, 2010.

[YWK+13] Xiaoyuan Yu, Jiangping Wang, Roland Kays, Patrick a Jansen, Tianjiang Wang, and Thomas Huang. Automated Identification of Animal Species in Camera Trap Images. *EURASIP Journal on Image and Video Processing*, 2013(52):1–10, 2013.

[YYGH09] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramidmatching using sparse coding for image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1794–1801, Miami, Florida, USA, 2009.

[YZ10] Meng Yang and Lei Zhang. Gabor feature based sparse representation for face recognition with gabor occlusion dictionary. In *European Conference on Computer Vision (ECCV)*, 2010.

[YZY11] M Yang, L Zhang, and Jian Yang. Robust sparse coding for face recognition. In *Computer and Pattern Recognition (CVPR)*, number 1, pages 625–632. Ieee, June 2011.

[ZCP03] W Zhao, R Chellappa, and PJ Phillips. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

[Zep13] Matthias Zeppelzauer. Automated Detection of Elephants in Wildlife Video. *EURASIP Journal on Image and Video Processing*, 46(1):1–44, 2013.

[ZL05] Cai-Nicolas Ziegler and Georg Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4-5):337–358, 2005.

[ZLZ10] Jiří Zuzaňák, Aleš Láník, and Pavel Zemčík. Description of image content by means of graph grammars. In *Poster Proc. 18th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2010)*, pages 43–48. Vaclav Skala - Union Agency, 2010.

[ZMM99] Ramin Zabih, Justin Miller, and Kevin Mai. A feature-based algorithm for detecting and classifying production effects. *Multimedia systems*, 7(2):119–128, 1999.

[ZMWZ00] Changqing Zhang, Weiyi Meng, Z Wu, and Zhongfei Zhang. WebSSQL - A Query Language for Multimedia Web Documents. In Michael P Papazoglou and Amith Sheth, editors, *IEEE Advances in Digital Libraries 2000 - ADL2000*, page 10. IEEE Computer Society, 2000.

[ZREK11] Frederik Zilly, Christian Riechert, Peter Eisert, and Peter Kauff. Semantic kernels binarized - a feature descriptor for fast and robust matching. In *2011 Conference for Visual Media Production (CVMP)*, pages 39–48, 2011.

[ZSG+05] Wenchao Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang. Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 786–791. Ieee, 2005.

[ZST08] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat Head Detection - How to Effectively Exploit Shape and Texture Features. In *European Conference on Computer Vision (ECCV)*, pages 802–816, Marseille, France, 2008. Springer.

[ZST11] Weiwei Zhang, Jian Sun, and Xiaoou Tang. From Tiger to Panda: Animal Head Detection. *IEEE Transactions on Image Processing*, 20(6):1696–1708, June 2011.

[ZW94] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision (ECCV)*, pages 151–158, 1994.

[ZZ10] Cha Zhang and Zhengyou Zhang. A survey of recent advances in face detection, 2010.

[ZZC01] Da-Qian Zhang, Kang Zhang, and Jiannong Cao. A context-sensitive graph grammar formalism for the specification of visual languages. *The Computer Journal*, 44(3):186–200, 2001.

ISBN 978-3-902448-43-9

MICO Early Adopters

MICO unites leading research institutions from the information extraction, semantic web, and multi-media area with industry leaders in the media sector.

Salzburg Research
Coordinator, Austria

Fraunhofer
Germany

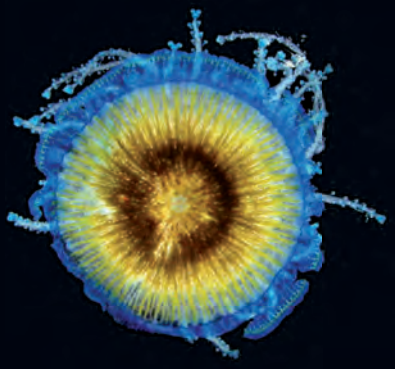
Insideout10
Italy

UMEA University
Sweden

University of Oxford
United Kingdom

University of Passau
Germany

Zaizi Ltd
United Kingdom



MICO is a research project partially funded by the European Union 7th Framework Programme (grant agreement no: 610480).