TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Mikko Roininen

Multimodal Video Analysis and Modeling

Julkaisu 1433 • Publication 1433

Tampere 2016

Tampereen teknillinen yliopisto. Julkaisu 1433
Tampere University of Technology. Publication 1433

Mikko Roininen

Multimodal Video Analysis and Modeling

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 18th of November 2016, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2016

ISBN 978-952-15-3845-2 (printed)
ISBN 978-952-15-3888-9 (PDF)
ISSN 1459-2045

Multimodal Video Analysis and Modeling

Mikko Roininen
Tampere University of Technology
Faculty of Computing and Electrical Engineering

October 25, 2016

Abstract

From recalling long forgotten experiences based on a familiar scent or on a piece of music, to lip reading aided conversation in noisy environments or travel sickness caused by mismatch of the signals from vision and the vestibular system, the human perception manifests countless examples of subtle and effortless joint adoption of the multiple senses provided to us by evolution. Emulating such multisensory (or *multimodal*, i.e., comprising multiple types of input modes or *modalities*) processing computationally offers tools for more effective, efficient, or robust accomplishment of many multimedia tasks using evidence from the multiple input modalities. Information from the modalities can also be analyzed for patterns and connections across them, opening up interesting applications not feasible with a single modality, such as prediction of some aspects of one modality based on another. In this dissertation, multimodal analysis techniques are applied to selected video tasks with accompanying modalities. More specifically, all the tasks involve some type of analysis of videos recorded by non-professional videographers using mobile devices.

Fusion of information from multiple modalities is applied to recording environment classification from video and audio as well as to sport type classification from a set of multi-device videos, corresponding audio, and recording device motion sensor data. The environment classification combines support vector machine (SVM) classifiers trained on various global visual low-level features with audio event histogram based environment classification using k nearest neighbors (k-NN). Rule-based fusion schemes with genetic algorithm (GA)-optimized modality weights are compared to training a SVM classifier to perform the multimodal fusion. A comprehensive selection of fusion strategies is compared for the task of classifying the sport type of a set of recordings from a common event. These include fusion prior to, simultaneously with, and after classification; various approaches for using modality quality estimates; and fusing soft confidence scores as well as crisp single-class predictions. Additionally, different strategies are examined for aggregating the decisions of single videos to a collective prediction from the set of videos recorded concurrently with multiple devices. In both tasks multimodal analysis shows clear advantage over separate classification of the modalities.

Another part of the work investigates cross-modal pattern analysis and audio-based video editing. This study examines the feasibility of automatically timing shot cuts of multi-camera concert recordings according to music-related cutting patterns learnt from professional concert videos. Cut timing is a crucial part of automated creation of multi-camera mashups, where shots from multiple recording devices from a common event are alternated with the aim at mimicing a professionally produced video. In the framework, separate statistical models are formed for typical patterns of beat-quantized cuts in short segments, differences in beats between consecutive cuts, and relative deviation of cuts from exact beat times. Based on music meter and audio change point analysis of a new

recording, the models can be used for synthesizing cut times. In a user study the proposed framework clearly outperforms a baseline automatic method with comparably advanced audio analysis and wins 48.2 % of comparisons against hand-edited videos.

Preface

The research work described in this dissertation was carried out at Tampere University of Technology (TUT) between 2010 and 2016. I would like to express my sincere gratitude to my supervisor prof. Moncef Gabbouj for all the guidance and "behind-the-scenes" arrangements, and to Anssi Klapuri for introducing me to (audio) signal processing work in the first place. I would also like to thank Miska Hannuksela, Igor Curcio, Antti Eronen, Arto Lehtiniemi, and Francesco Cricri for offering me interesting projects at Nokia Technologies Labs as well as at the former Nokia Research Center. Additionally, I would like to thank Esin Guldogan, Michal Joachimiak, Junsheng Fu, Toni Mäkinen, Ugur Kart, Sujeet Mate, Jussi Leppänen, as well as all members of the Audio Research Group and MUVIS group that I've had the pleasure to work with over the years. Finally, I want to thank my wife Johanna, my son Aleksi, my sister Elina, and my parents Aulikki and Jarmo.

Contents

Acronyms

ANN	artificial neural network
CCA	canonical correlation analysis
CNN	convolutional neural network
DBN	dynamic Bayesian network
DCT	discrete cosine transform
DS	Dempster-Shafer
EKF	extended Kalman filter
GA	genetic algorithm
GLCM	gray-level co-occurence matrix
GMM	Gaussian mixture model
GPS	Global Positioning System
fMRI	functional magnetic resonance imaging
HDR	high-dynamic-range
HMM	hidden Markov model
HSV	hue, saturation, value
KF	Kalman filter
k**-NN**	k nearest neighbors
LDA	linear discriminant analysis
LSI	latent semantic indexing
LSTM	long short-term memory
MC	Markov chain
MFCC	Mel-frequency cepstral coefficient
MKL	multiple kernel learning
MLP	multilayer perceptron

MR-CCA multiple ranking canonical correlation analysis

ORDC ordinal co-occurrence matrix

PCA principal component analysis

PF particle filter

RBF radial basis function

RBM restricted Boltzmann machine

R-CCA ranking canonical correlation analysis

RMS root mean square

RNN recurrent neural network

STIP space-time interest points

SVM support vector machine

UKF unscented Kalman filter

List of Publications

1. Mikko Roininen, Esin Guldogan, Moncef Gabbouj, "Audiovisual video context recognition using SVM and genetic algorithm fusion rule weighting," *in Content-Based Multimedia Indexing (CBMI), 2011 9th International Workshop on*, pp. 175 – 180 Jun. 2011.

2. Francesco Cricri, Mikko Roininen, Sujeet Mate, Jussi Leppänen, Igor D. D. Curcio, Moncef Gabbouj, "Multi-sensor fusion for sport genre classification of user generated mobile videos," *in Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pp.1-6, 15-19 July 2013.

3. Francesco Cricri, Mikko Roininen, Jussi Leppänen, Sujeet Mate, Igor D. D. Curcio, Stefan Uhlmann, Moncef Gabbouj, "Sport Type Classification of Mobile Videos," *in Multimedia, IEEE Transactions on*, vol.16, no.4, pp.917-932, June 2014.

4. Mikko Roininen, Jussi Leppänen, Antti J. Eronen, Igor D. D. Curcio, Moncef Gabbouj, "Modeling the timing of cuts in automatic editing of concert videos," *in Multimedia Tools and Applications*, 2016, DOI 10.1007/s11042-016-3304-7.

1 Introduction

Video is a rich information medium with a wide abstraction gap between the low-level signal representation and the semantic content captured therein. While the human brain can effortlessly decode the visual stimuli received by the eyes into the semantics of the perceived scene, modeling this process by a computer even partially in a carefully constrained scenario is far from trivial. The process of computationally extracting higher-level semantic information from a sensory signal is commonly known as automatic content analysis. An example of a video content analysis task would be the detection and tracking of a moving face in a video clip. The automatic content analysis algorithm needs to identify patterns and regularities in the input signal, infer various commonalities between the patterns, and group the patterns accordingly.

Automatic analysis of professionally produced, edited, or otherwise post-processed video material has been studied extensively for some decades (see, e.g., [1–3] and their references). With the democratization of video authoring and broadcasting due to the proliferation of affordable and easy to use recording, storing, and sharing tools and services, the focus of attention of the video content analysis research community has increasingly widened to cover these user-generated recordings besides mere traditional professional content. Although established solutions exist for many low-level problems affecting user-generated content (e.g., stabilization, auto-focus, automatic gain control) either due to long-standing related research or the fact that methods can be easily adapted from the professional domain, semantically more abstract tasks could also benefit from automatization. The automatization of more abstract tasks is less straightforward as there are larger variations and less quantifiable aspects in such higher-level tasks. Besides the on average lower and more highly varied overall quality of user-recorded videos, such content generally has less structure compared to professionally produced videos due to lack of editing, and rarely contains information augmented in post-processing. Due to the massive amount of potential content creators, user-generated content has the advantage of much more comprehensive coverage of various public events of different sizes as well as private events that are rarely documented with professional equipment. However, this also means that user-generated content is created in considerably higher volumes. This along with the varied quality of the content prompts for increased focus on automatic content analysis and processing of such content.

The practical applications of automatic video content analysis are multitudinous including fields and tasks such as video database indexing and retrieval, video summarization, human-machine interaction, surveillance and scene understanding, biometrics identification, affective analysis, augmented reality, medical analysis and monitoring, assisted living, attention and saliency modeling, sports performance analysis and automatic statistics generation, source separation, music content analysis, automatic or assisted video production and editing, as well as scene-aware exploration and navigation of autonomous vehicles

and robots [4–8]. Common problem types encountered in many of the aforementioned applications as well as other multimedia analysis tasks include segmentation, event detection, structuring, and classification [4]. Segmentation is the spatial or temporal splitting of the multimedia item. Event detection deals with identifying specific discrete incidents. Structuring deals with full and often hierarchical segmentation of a multimedia item as well as identifying or labeling the segments. Classification assigns entities to named categories and is used for various labeling and identification subtasks. The first three tasks often deal with temporal data, which needs to be taken into account accordingly in the processing.

The automatic content analysis tasks are made more challenging by the fact that often the sensed objects of interest have many degrees of freedom, which results in large variations in the sensed signals within a semantic grouping. In many automatic content analysis tasks the natural semantic groupings of the sensed patterns might actually be practically impossible to achieve by linear comparisons in the input space: in the input space an image centered on a white cat is likely to resemble an image of a white dog more than an image of a black cat. Similarly, the sensed audio signal of a middle C note played on a piano might in some sense resemble much more the middle D note on a piano than the middle C on a guitar. Further complexity is added by various sources of noise and variation, such as environment conditions (e.g., lighting, background sounds) and variations in the relation between the sensor and the target (e.g., target movemement, sensor movement, sensing position, sensing direction, occlusions). To tackle these issues, the sensed signal should be transformed into a form, where unrelevant information is attenuated while keeping semantically relevant aspects and representing them so that grouping and comparison becomes easier.

Automatically understanding unconstrained everyday scenes and extracting knowledge into compact models from high-resolution and high-field-of-view (up to full 360 degree) videos at high-enough spatio-temporal fidelity reliably yet efficiently is still largely an unsolved problem. The efficiency aspect becomes increasingly important with increased visual data acquiring rates due to increasing resolutions, higher frame rates, multiple streams needed for 3D video, or capturing a scene with multiple cameras. Multi-camera setups can be used for instance for increased field of view (without sacrificing resolution) by stitching the views from a fixed camera array, or for recording an event or scene from different viewing angles by multiple devices.

One way of achieving improved efficiency in certain video analysis tasks is by utilizing multimodal analysis, which is the process of intelligently combining information from different modalities, i.e., sensors or sources of different type such as audio and video. The use of multiple complementary modalities has potential for improved performance, efficiency, and robustness to external conditions, such as noise in a single modality. As an example, excluding the effect of certain camera operations, such as panning and tilting, in object motion analysis based only on the video content requires advanced motion analysis and is error-prone to movement in the target scene. Yet, the required camera motion information can be acquired from motion and orientation sensors with practically no additional computation and without the ambiguity between camera and target motion. Combining information from multiple sensors or modalities can thus allow satisfactory levels of performance on a task with increased cost-efficiency compared to spending resources trying to improve a single sensor [9]. Another use case of multimodal analysis is to find common patterns and dependencies between the modalities for cross-modal inference.

The scope of the dissertation is limited to the multimodal analysis of sequential data streams, such as video, audio, and various auxiliary sensors. Additionally, in all the experiments described in the thesis, some part of the processing is done for unedited non-professional video content. Specifically, professionally recorded and edited video material is only used for modeling concert video cut timing patterns.

1.1 Machine learning

Machine learning is a valuable tool for contemporary automatic content analysis. Machine learning aims at identifying relations and regularities in example data. A successful machine learning algorithm is able to produce desirable output for unseen data by utilizing the knowledge gained from the already seen examples. This is known as generalization.

Machine learning approaches can be categorized by their use of human supervision. In supervised learning the algorithm is provided with the correct response (e.g., discrete class label for classification or continuous output value for regression) corresponding to each training example. The aim is then to learn a mapping that generalizes to unseen data. The major drawback of supervised learning is the manual annotation needed for labeling the training data. The unsupervised learning problem lacks the correct responses provided in supervised learning. In unsupervised learning logical structure needs to be extracted by inspecting the example data alone without any correct labels. One example of unsupervised learning is clustering, where the data is split into coherent groups. The difference to supervised classification is that the clusters are not typically assigned with any meaningful identity information other than what can be extracted, e.g., from their sizes or their distributions in the feature space. Semisupervised learning aims at combining the advantages of supervised and unsupervised learning. Namely, the learning algorithm is provided with a combination of few labeled and many unlabeled examples. The idea is to utilize the fact that both types of examples have been generated from the same distribution, and thus mapping between the unlabeled and labeled data can be estimated and the unlabeled data be used to refine the model fitting to the supervised data. In active learning the learning algorithm queries a human expert for correct responses to chosen examples. While efficient active learning should focus the queries only to the difficult cases, e.g., near the class borders for classification, scalability can easily become an issue with relatively slow human input in the loop. In reinforcement learning the supervision comes in the form of a reward signal instead of providing the correct answers. The aim of the learning is to maximize the long-term reward. Transfer learning or multitask learning tries to utilize the knowledge gained from a supervised learning task to learn another, related task. In multimodal context, a specific set of machine learning algorithms, so called cross-modal learning uses one modality as supervising signal to another modality. In this thesis only supervised and unsupervised learning are considered.

Machine learning model selection and optimization needs to balance between errors introduced by so called *bias* and *variance* [10]. Bias arises from the utilization of overly simple models with limited amount of degrees of freedom. Variance increases with overly complex models. High-variance models have the capacity to match the input data precisely. However, in the process they can also capture unrelevant details of the training set, which prevent them from generalizing well to unseen data. This problem is known as overfitting. Correspondingly, models with high bias fail to capture some relevant aspects of the training data leading to so called underfitting, i.e., bad generalization due to not learning from the training data all the properties relevant to a given task.

As bias and variance are related to low- and high-complexity models, respectively, there's always a tradeoff between them for a given data set. Techniques or modeling choices with high bias and low variance systematically produce relatively similar results that fail to take into account some relevant aspects of the problem and thus differ prominently from the optimal solution. In contrast, high-variance low-bias methods vary significantly between different data sets sampled from a common distribution, but produce good modeling as averaged over the different samples. However, generally in practical problems only a single data sample from the generating distribution is available. Thus any single training procedure run with fixed parameter values will have unpredictable exact contributions from bias and variance to the resulting generalization error.

A common performance metric in classification is the correct classification rate or classification accuracy:

$$\mathrm{acc} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(y_i, t_i), \tag{1.1}$$

where N is the evaluation data set size, y_i and t_i respectively the prediction and target value for sample i, and $\mathbf{1}(\cdot,\cdot)$ the indicator function returning 1 if the arguments are equal and 0 otherwise. Although classification accuracy is a simple and intuitive measure of performance, it can give biased performance estimates for instance with unbalanced datasets. As an example, if an evaluation data set of a binary classification problem consists of 95 samples of one class and 5 samples of another, a trivial classifier that blindly predicts the first class regardless of the input gets accuracy of 0.95.

Parameter optimization for balancing the bias–variance tradeoff in hopes of low generalization error is typically done by applying different sampling techniques to the input data set[1] provided for training the system. Common such techniques include holdout, cross-validation, and bootstrap sampling [11]. In holdout the input data is partitioned into mutually exclusive training and testing set. In k-fold cross-validation the input data is randomly split into k subsets, each of which is used for evaluating a model trained on all the remaining data. The overall estimate is then acquired as the average over the folds. A special case of k-fold cross-validation is the leave-one-out cross-validation, where k is chosen as the amount of samples in the input data, i.e., each fold is evaluated on a single input data point and trained on all the rest. Sometimes it is advisable to form the folds according to some naturally occurring structure in the data. As an example in one of the experiments in chapter 2, videos have been recorded in different sport events by multiple people, so a fold is defined as all the videos from a specific sport event, as their content is likely to be more correlated with each other than with videos from other events at different locations and time. Bootstrap sampling set is formed by drawing (with replacement) as many samples as is the amount of samples in the input set. All samples in the boostrap sample are then used for training and the rest for evaluation. In stratified sampling the sample drawing is constrained so that the relative proportions of different labels in the sample are roughly equal to the proportions in the input data set. This can be advantageous especially in imbalanced problems, where the example amounts between different classes in the input data set differ considerably. Non-stratified sampling might in such cases result in samples without any instances of a minority class. The sampling procedures can be repeated multiple times for more reliable parameter

[1] Distinction is here made to the term *training data set*, which is used to refer to the portion of the input data that is shown as examples to the model induction or training algorithm.

optimization and performance evaluation. This, however, increases the overall required training time roughly linearly.

If model induction or other parameter optimization is performed in an iterative manner, the paramater choices at every iteration should be evaluated on a separate validation data set, and only the final performance evaluation done on the test set that has not been used during the optimization process. The main reason behind this is that if the testing data are allowed to even indirectly affect any choices in the modeling, the model will be biased towards the distribution of the testing data and will produce overly optimistic estimates for the system performance on truly unseen data.

Classifier stability affects their applicability in multi-classifier systems. A stable classifier, such as a SVM, tends to produce a similar decision boundary regardless of small changes introduced to the initialization of the training, such as the order at which the examples are fed to the classifier or how the possible internal state of the classifier is initialized. With unstable classifiers, such as multilayer perceptrons (MLPs) or decision trees, small perturbations in the initialization may result in large changes in the learned model. Unstable classifiers are suitable for multi-classifier learning within a single modality as it is often easier to create multiple diverse classifiers with unstable algorithms compared to stable methods. In multimodal learning, diversity and complementarity are inherently present due to the different nature of the modalities, so the degree of stability of the learning algorithm generally affects the performance less.

The representation, in which the data is fed to the machine learning algorithm, often plays a key role in the success (or failure) of the learning task. Feature extraction – and all manual or automatic hierarchical representation refinement in general – aims at transforming the data to retain the essential information while suppressing irrelevant noise and compressing the data amount. The compression in dimensionality can also aid learning algorithms to avoid overfitting, which is the process of over-optimizing the model to fit irrelevant intricacies in the training data. Another aim is to transform the data into more suitable form for a given task (e.g., class-wise more easily separable form in case of classification). From a multimodal viewpoint, modality representation refinement can alleviate the so called incommensurability problem, i.e., the mismatch between modality representations due to heterogeneity in physical units, value range, resolution, dimensionality, and tensor order of the data [12]. In sequential tasks, one common way of augmenting temporal information to stationary features extracted at discrete points of sequential data, is to calculate the first and second order derivatives (in practice usually discrete differences) of the sequence of stationary features [8]. Alternatively, specific modeling tools with properties for implicitly taking the temporal aspects into account can be used. Popular such approaches include long short-term memory (LSTM) [13] and other recurrent neural networks (RNNs) as well as hidden Markov models (HMMs).

Recently, in many multimedia content analysis tasks, one major trend has been to make the representation refinement process more automatic and data-driven. Data-driven end-to-end systems alleviate the need for explicitly engineering the feature extraction logic of the inference process – yet often considerable experimentation is still needed for defining the optimal architecture for the end-to-end learning. Especially with approaches capable of handling vast amounts of data, the implicit data refinement of such end-to-end systems has recently been shown to surpass hand-engineered solutions in domains such as visual object recognition and localization and speech recognition. However, the automation of the representation optimization can make the inner workings of the learning system less transparent and more difficult to understand.

1.2 Multimodal analysis

Essid and Richard [7] distinguish between two main types of multimodal analysis tasks: cross-modal processing and multimodal fusion. In the former, the task is to reveal various dependencies, relations, and common patterns between the different modalities with regard to the analyzed content, e.g., for cross-modal prediction, whereas in the latter the aim is to gain advantage for completing an analysis task more effectively by utilizing the joint information of the modalities. In a more specific and architecture-constrained categorization Ngiam *et al.* [14] describe the distinct tasks of multimodal fusion, cross-modality learning, and shared representation learning. The grouping is based on the availability of modalities at different stages of their representation learning pipeline. In multimodal fusion all modalities are available during unsupervised representation learning, supervised task training, as well as the operation phase of the trained system. In cross-modality learning multiple modalities are used for the representation learning phase with the aim of obtaining improvements for a unimodal supervised task, i.e., using one common modality for supervised training and testing. In shared representation learning the representation learning phase is again multimodal, and single modalities are used for training and testing, but in contrast to cross-modality learning the training and testing modalities are different. This dissertation follows the more generic categorization of [7]. Accordingly, publications 1, 2, and 3 (discussed in chapter 2) utilize multiple modalities in a multimodal fusion manner, whereas publication 4 (discussed in chapter 3) considers cross-modal processing. Hence, cross-modal processing is presented in more detail in chapter 3 and the remainder of this overview section concentrates on multimodal fusion. Figure 1.1 shows an overview of the multimodal analysis concepts considered within the thesis.

In multimodal analysis literature the terms *modality* and *multimodal* have various definitions. Jaimes and Sebe [15] define modalities as corresponding to different senses or input devices. Thus, according to this definition, for instance combining hand gesture and facial expression recognition from video is not multimodal as both information sources are analyzed from the same sensor. In [16] the term modality is used for any specific information acquisition framework, such as different types of detectors used at different conditions, different observation times, or in multiple experiments or subjects. In this dissertation the term modality is used rather loosely to refer to any separate information sources – including different features extracted from a common sensor – combined or coanalyzed in a multimodal context. The terminology for the utilization of a single modality also varies between authors from uni-modal or unimodal [16, 17] and monomodal [5, 7] to single-modal [12].

Lahat *et al.* [12] list motivations for data fusion in multimodal context: combining multiple data sources about a system of interest allows broadening the view for a more complete understanding of the system. Additionally, multiple data sources may also allow improved decision making, exploratory modality relationship research, question answering about the system, and knowledge extraction in general. Besides these arguments, an intuitive way of motivating experimentation on multimodal analysis is to think of the human sensory information processing chain, i.e., sensation, perception, and cognition [4]. Evolution has armed us with multiple senses that respond to distinct types of stimuli at the sensation level. At the perception level the sensed information is filtered, selected, organized, combined, and interpreted. Finally, at the cognition levels the multimodal information from the perception level is further refined and analyzed to accomplish tasks such as comprehension, learning, memorizing, decision-making, and planning. Humans

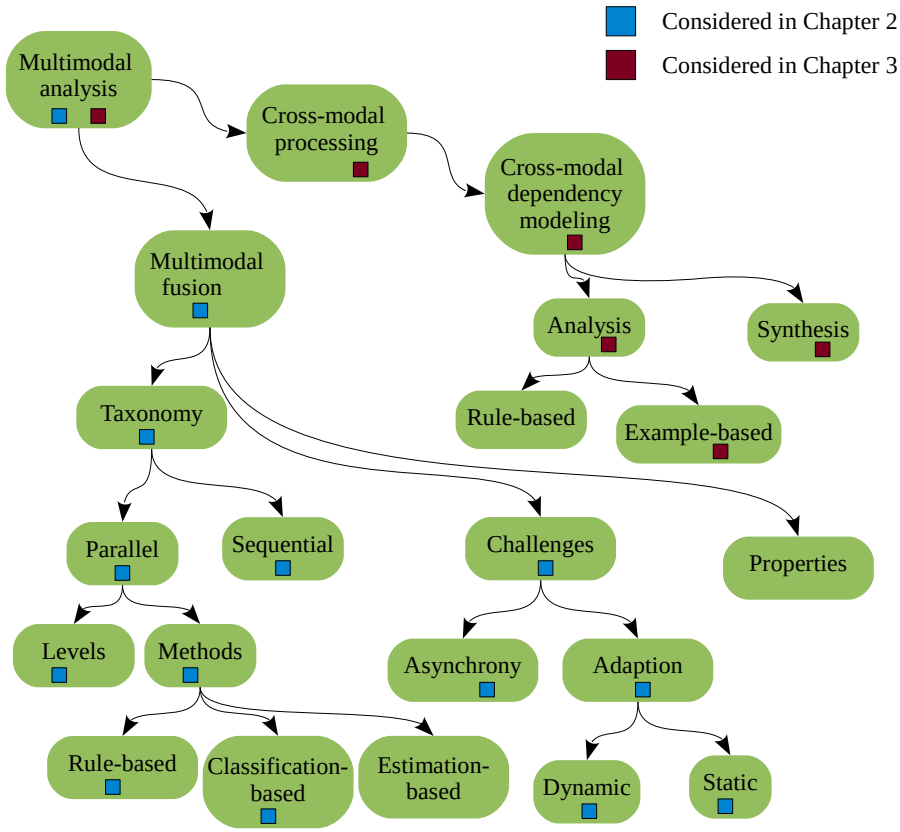

Figure 1.1: Overview of the taxonomy of concepts considered in the thesis. Color-coded squares indicate, in which chapters the concepts have been utilized.

utilize this chain effortlessly with the different senses to interact with a highly dynamic and evolving environment [12]. Machine learning can be seen as means for approximate inversion of the mapping of the world state to the sensed signals, i.e., in case of multimedia tasks essentially simulating the perception and cognition processes [4]. Many approaches to multimodal fusion (as well as to intelligent systems in general) have thus been drawing inspiration from human cognition and more generally natural processes. Even complete subfields of computer science and optimization, such as evolutionary computation, are based on studies of adapting natural phenomena to computational problems [18]. However, it is not always purposeful to limit the design principles too strictly to simulating nature [6]. A common counterargument for strictly imitating nature in technological development is that of the aeroplane: even though the concept of wings has been inspired by birds, taking off and maintaining the airspeed of aeroplanes are enabled by entirely different means than flapping the wings.

Naturally, multimodal analysis generally comes with increased complexity compared to unimodal approaches as the amount of data streams increases and the combination process further adds to the complexity. However, this can usually be justified by the

gains in robustness and performance. In many cases multimodal analysis offers a way for surpassing the unimodal performance upper bounds on a given task [9]. Even though the unimodal performance can be optimized with new data or the systems further tuned with domain-specific contextual information, and often seemingly stagnant performance on a task is suddenly pushed with a novel breakthrough approach, some conditions or other challenges may still affect a modality in such a fundamental way that no methodological changes can prevent a systematic failure. A simple example is trying to carry out visual-only recognition tasks in conditions too dark for the imaging sensor. Besides, the extra information from an added modality might actually simplify a problem, so that much simpler models are required compared to the unimodal case.

1.2.1 Relation to other information fusion approaches

Multimodal fusion is also closely related to multiview learning, where data from multiple distinct views is combined for improved or more robust learning [19]. For instance, separate data sets describing a common domain or different feature subsets can be considered as different views [17]. Unlike the strict definition of multimodality, in multiview learning the different views can originate from a common modality, for instance by using different feature extraction methods. However, the exact definitions vary between authors and the restrictions are not always strictly followed. Thus, the terms are sometimes used interchangeably. In [4] the term *multicue* is used for approaches that combine multiple distinct representations from a single modality to distinguish them from multimodal fusion.

Advantages of combining multiple decision making entities have been studied in the field of ensemble learning, which considers the combination of the decisions from multiple sources in a way that surpasses the performance of the individual component decisions. More specifically, the methods aim at reducing the overall variance by the combination process. In traditional ensemble learning usually a single modality is used and the different decision making entities are derived for example by subsampling the data, with different initialization conditions for the learning, or using different learning algorithms.

In generic data and sensor fusion, considerable amount of literature exists for the combination of data from multiple similar sensors both with fixed and unconstrained spatial arrangements. In contrast to multimodal fusion, these methods have the advantage that the different sources are usually in the same representation and synchronized (or easily synchronizable). Yet, they share relatively similar view of the problem of interest, which renders them sensitive to similar disturbances. Output of such systems – such as sound direction of arrival estimate from a microphone array – are commonly used as a single modality for multimodal analysis. Lewis and Powers [20] use the term competitive data fusion to distinguish it from complementary data fusion. In the former case the fusion is done between multiple similar information sources in the hope of increased overall performance by exploiting the lack of correlation in their errors or other noise. By complementary data fusion the authors mean the utilization of multiple diverse sensors or other means for having a distinctively different view of the system of interest between the sensors, e.g., multimodal or multiview learning.

1.2.2 Elements of robust multimodal fusion

As with any information fusion, in order to be of any value the fused sources should bring some unique additional information to the whole, i.e., they should complement each other.

In many cases combining multiple distinct data modalities gives notable complementarity as the different information sources sense entirely different properties of the target scene. An ideal multimodal system should be able to dynamically balance the contributions of each modality in an optimal way based on their data quality, momentary reliability, and confidence in their decisions [8]. The confidence should also depend on various sources of contextual information as well as the input data properties, and additionally the fusion should be robust to imperfections in the input, e.g., environment and sensor noise as well as missing data [6, 15]. The individual modalities should be transformable to a joint representation space, where their dependencies relevant to a given task can be easily exploited [15]. Similarity in the joint representation space should correspond to similarity of the high-level concepts for intuitive classification and retrieval, and obtaining the joint representation should be easy even with missing modalities or values [21]. Lahat *et al.* [12] point out that in order to develop domain-free, widely applicable fusion methods, they should be data-driven and utilize only weak priors and constraints, such as sparsity, nonnegativity, low-rank, independence, smoothness, etc. This is in accordance with the prevalent trend towards data-driven end-to-end systems.

Various partially unsolved challenges have been reported for multimodal fusion in the literature including: optimal utilization of correlation, independence, contextual information and modality confidence, synchronization between modalities, optimal modality selection, optimizing complementarity of modality representations and models, combining different units, dimensionalities, tensor orders, and temporal and spatial resolutions of modalities, dealing with noise/conflicts/inconsistency, and handling missing data [4, 5, 15, 16]. Many of the challenges are directly related to the presented desired properties, and are reported as challenges as their utilization is still far from optimal or solutions exist only to strictly limited domains. The above mentioned properties and challenges are discussed in detail in the following sections.

Uncertainty

Multimodal fusion should be robust to various sources of uncertainty. One typical source is any interference added to or otherwise intertwined with the information from the sources of interest: calibration errors, finite precision, quantization or other quality degradation, or noise from environmental conditions (e.g., thermal noise, reverberation, ambient noise, visual distortions due to unfavorable light conditions) [12]. Poh *et al.* [9] distinguish between sensor, channel, and modality-specific noise in the sensed data. Sensor noise results from the measurement imprecisions of the sensor. Channel noise is the interferences introduced while the sensed information is transmitted between the target and the sensor. This includes for instance environmental noise and nonoptimal lighting. The modality-specific noise arises from deviations from any assumptions or constraints set for a modality, such as occlusions or unfavorable sensing directions (e.g., head pose variations, when assuming a frontal face for person identification). Noise can be attenuated with different filtering and smoothing techniques, which are often based on various assumptions about the data such as smoothness, e.g., in the temporal or spatial dimensions, or lack of correlation between the noise of multiple similar sensors. Another approach – applicable also to data with no temporal or spatial relations within or between data points – is trying to identify and completely remove the noisy data samples prior to the processing [5]. Most methods for attenuating output noise by combining information from multiple sources, assume that the noise of each individual source is independent of the noise of the other sources. This assumption might not be satisfied,

which might lead to bias as correlation in the noise between the sources is interpreted as the signal of interest [12]. However, if the different information sources are heterogeneous modalities, it is generally more likely that the noises are less correlated as well. Thus, complementary modalities can also reduce the effect of noise in one modality to the overall system performance.

Another common source of uncertainty is missing data [8, 12]. Lahat *et al.* [12] list various reasons for missing data: unavailability, unreliability, or discarding of data entries due to faulty detectors, occlusions, partial coverage, or other effects; modality sensing range limitations or other partial coverage compared to other modalities; combining modalities with partially common dimensions and interpreting the non-intersecting dimensions as missing values; as well as interpreting a lower-resolution modality as having missing data at the sampling points of a more densely sampled modality. Data can be missing systematically or spuriously from single feature elements or complete modalities might be unavailable. In some cases some modalities can be available at the training phase, but might be missing at testing or operation time [8].

Asynchrony

Different modalities have their optimal ranges of sensing rates, which are determined by task- and modality-specific constraints – such as the Nyquist limit for the highest frequency reconstructable with a given sampling rate – as well as by sensor and processing chain capabilities. The rate can range from constant (e.g., audio and video sampling rates) to very sparse and sporadic, such as in the case of keyword spotting from speech or text. In addition to the asynchrony from different data acquiring and processing rates, in certain tasks, the modalities might have a natural asynchrony in their information content. For instance, in audiovisual speech recognition the mouth shape corresponding to a specific acoustic phone may start notably before or end notably after the occurrence of the phone. Katsaggelos *et al.* [8] exemplify this phenomenon: When pronouncing the word "school", the lips typically begin to round for the /uw/ sound while still producing the sound /k/ or even /s/, which is an example of so called anticipatory coarticulation. Correspondingly, in preservatory coarticulation the mouth gesture continues after the sound has already stopped or changed.

The synchronization needs are also affected by the abstraction level, at which the modalities are to be fused in the processing chain [5]. Specifically, often the synchronization needs can be relaxed by fusing the modalities at higher abstraction levels. This is especially true if aggregating data within temporal windows to higher-level information with sparser granularity. Two commonly used synchronization methods are: taking the newest most recent data at each modality at regular intervals, or waiting until new data is available from all modalities [5]. The modalities may also require different minimum amounts of consecutive data to accomplish a given task, e.g., detecting a person walking in video as opposed to detecting the sound of footsteps from audio [5]. Similarly, the effective completion of different tasks in a single modality may require highly different amounts of data [8]. For example, the presence of a person can generally be detected from a single image or video frame, whereas recognizing their current action requires in many cases the analysis of a longer video segment. The effective data amount for carrying out a task in turn affects the granularity, by which results from this task can be output for higher-level tasks. The optimal granularity for a given modality in a given task is often a tradeoff between high output rate and the confidence of the decisions. Snoek and Worring [22] argue that often the choice of a certain level of granularity over all modalities is based on

the natural granularity of the main modality of expertise or preference of the researchers. Especially with increased amount of modalities, sometimes it is more efficient to choose a level of granularity somewhere between the unimodal granularities. This issue of choice is not unique to temporal synchronization, but concerns also, e.g., spatial resolution and other differences between the modality representations. Different asynchrony sources result in various degrees of asynchrony and the degrees of multiple sources may accumulate. This needs to be taken into account in the fusion process. The synchronization issues are more critical to online systems and often ignored in the literature due to commonly used offline experimentation [5]. Yet, the utilization of multiple modalities with asynchronous data sampling rates or phases can result in increased or more consistent overall rate of obtaining data as new evidence from different modalities is received at different times [8].

A related problem, alignment to a common coordinate system, can be thought of as a two- (e.g., 2D image coordinates) or higher-dimensional (e.g., 3D world coordinates) analogy to synchronization. Different modalities sensing with a common dimensionality might have misalignments between their coordinate systems due to different types of noise in the information the modality measures, spatial distortions and differences – such as different fields of view, varying contrasts, and misalignment of the sensor positions in relation to the target scene [12].

Incommensurability

Besides asynchrony different modalities may have other heterogeneities between their representations that complicate or even prevent direct comparison and matching. This issue is known as incommensurability or noncommensurability [12]. Such heterogeneities include but are not limited to the different physical units measured by the modalities; different value ranges or distributions; different spatial, temporal, or spectral resolutions; incompatible differences in data amounts; different orders of the representations, e.g., vectors, matrices, and higher-order tensors; and different dimensionalities within a common order [9, 12].

To alleviate the problem, the modalities can be transformed into more compatible representations. Depending on the task and need, specific transformations can be utilized for obtaining similar properties in some or all aspects of heterogeneity. It might for example suffice to transform the modalities into representations with the same order and value ranges but different dimensionalities for concatenating the representations into a higher-dimensional multimodal representation for further processing. Fully matching the representations between modalities has the advantage of enabling direct comparison as well as cross-modal inference. Analogous to the choice of granularity, the common representation can be chosen among the representations of the modalities, as a combination of different heterogeneities from different modalities (e.g., using the spatial resolution of one modality but the value range of another), or as a completely new, latent representation. In the last case, choosing a representation in between those of the modalities might minimize the extremity of the needed transformations, which would be both efficient and balance the information distortion among the modalities. However, emphasizing simpler and less granular representations could also boost efficiency, and on the other hand sometimes additional complexity might improve performance, as exemplified by kernel methods such as SVM. In many cases the latent representation is learnt algorithmically rather than chosen manually. Yet, achieving the full matching might not always be practical or even possible due to too extreme loss of vital information in the transformation process, if the degree of incommensurability is too high. In some cases, obtaining multimodal

information in a common representation can be done by transforming the content into textual form. As an example, in [22] the outputs from optical character recognition and speech recognition are fused. This enables the use of established text matching and document retrieval methods such as latent semantic indexing (LSI) [23]. However, this conversion to text domain is only usable in a limited setting and ignores much of the content, so it rarely suffices to be used as the sole analysis method.

Redundancy, unimodal performance, and complementarity

Multiple modalities that measure or reflect the relevant aspects of a problem should ideally have dependencies between each other, which helps reducing the errors due to variance as discussed in 1.1. This gives robustness against noise or missing data in single modalities [8]. The redundancy can also reveal the unique solution to an otherwise unsolvable ambiguous problem [16]. Yet, it is argued in [17] that improper handling of redundant information might cause various types of overhead such as unnecessarily high dimensionality in the fused data. Intuitively the individual modalities should perform as well as possible in the target task. Yet, there is a tradeoff between good unimodal performance and the contribution brought by the modality to the fusion. With higher average performance, the modalities produce more and more redundant results. In the extreme case of identical predictions between two modalities, their fusion adds no value as similar results can be achieved with less overhead with a single modality.

One of the main aims for multimodal fusion is that the different modalities should aid each other to counter their individual weaknesses by providing complementary information. The term diversity has been used in the literature to describe the complementarity-providing differences among a set of modalities (or in a broader context multiple decision-making entities) [24, ch. 10]. Diverse modalities have correlated correct predictions, but uncorrelated errorenous predictions. Using multiple complementary modalities can also broaden the applicability of a task, such as extending traditional audio-based speech recognition with using visual information for speech recognition from lip reading of a mute person [9]. Complementarity may also relax the need for manual annotations as modalities can act as fuzzy labels to each other – one modality might be invariant to large variations in another [21]. As an example, two different words could be assumed related or even having a common meaning, if they are frequently used to describe the same images.

Context adaption

Modalities can be monitored for different aspects affecting their performance. These include the usage context (e.g., the location and time of day), the confidence of the decisions, and the quality of the data, such as the presence of a type of noise the modality is sensitive to. Atrey *et al.* [5] distinguish between environmental and situational context. The former includes aspects such as time, sensor location and orientation, geographical location, sensor parameter values, or weather, whereas the latter includes, e.g., user mood and identity. They also note that context can be obtained either by content analysis (e.g., mood estimation from voice) or with dedicated sensors (e.g., time, positioning). Both momentary contextual confidence of different modalities and their longer-term reliability in certain relatively fixed conditions should be taken into account for robust multimodal analysis.

To this end, it would be advantageous to adapt the fusion process in a way that the

modalities, which are more likely to result in the desired decision are emphasized over less confident or robust modalities. If the adaption is done once in an offline manner the adaption is called static [8]. In dynamic adaption, the fusion process is actively adjusted during the operation according to any contextual information, such as modality decision confidence or data quality. A typical way of adaption is to assign weights to the different modalities according to some measurable or computable criterion [5, 8, 9]. It is also possible to use the criterion signals as additional input to the fusion process (e.g., by concatenating them with the modality decisions) [9]. An ideal adaption criterion should correlate well with the modality performance on a task. Poh *et al.* [9] distinguish between feature-based criteria, where the quality of the information of the modality is measured (e.g., by signal-to-noise ratio [8]) and decision-based criteria, which estimate the decision reliability or confidence. They also point out that often it makes sense to combine multiple criteria that measure different sources of degradation or confusion of the modality information.

Further challenges and trends

Besides the challenges presented in this section, various other open issues have been reported in the literature over the past decade. Most of them are still relevant, remain largely unsolved, and handling them would contribute to the efficiency and robustness of multimodal analysis. One important issue is improving the utilization of larger amounts of novel modalities and more intelligent nonlinear and semantic level relation mining for cross-modal processing [5, 7]. Utilization of unlabeled data would be advantageous as in multimedia analysis context, data is usually much easier to obtain than to annotate even for simple descriptive information such as binary exclusive presence of a certain concept in the content. With more elaborate and complex labeling, such as precisely spatially or temporally locating possibly multiple instances of multiple object classes, the labeling becomes more and more laborous. Besides pure unsupervised learning approaches, unlabeled data can be used for instance with semi-supervised, transductive, and active learning [25]. Judging from the breakthroughs of the recent representation and end-to-end learning methods on single modalities, hierarchical, data-driven, modality-agnostic end-to-end approaches are expected to improve over domain-specific constrained procedures [8, 12]. However, most current methods of the former type require large quantities of labeled data to truly shine.

1.2.3 Taxonomy of multimodal analysis

Multimodal video analysis can be categorized by various aspects of the problem. These include but are not limited to the used modalities, the task domain, the level of temporal precision, the spatial analysis level, the degree of temporal synchronization between the modalities, the causality or realtime requirements of the processing, passive vs. active (including, e.g., interaction or people carrying sensors knowingly) analysis, or by computational or other cost differences [5] [6]. Ruta and Gabrys [26] distinguish between classifier fusion and dynamic classifier selection, where a single classifier is chosen among the component classifiers in a multi-classifier system on a per-sample basis during operation. This can be thought of as a special case of dynamic adaption with the choice of weights limited to the set $0, 1$, and only allowing a single classifier to obtain weight value of 1 at a time. Fusion and selection can also be alternated hierarchically on subgroups of the components. Multimodal fusion methods can also be categorized as parallel (i.e., simultaneous) or sequential (i.e., ordered) [16, 22, 26]. In parallel multimodal fusion,

information from all the modalities is combined at the same time for a fused decision. In sequential fusion multiple fusion methods can be applied in succession, or the different modalities can be used as a cascade to narrow down the set of possible classes until a single decision can be made with reasonable confidence. Another successive processing startegy is to use certain modalities to filter subsets or segments of the data to be fed to another modality for further analysis [4]. Besides simultaneous and ordered fusion, Snoek and Worring [22] assort fusion approaches according to the use of statistical versus knowledge-based classification and the processing cycle being iterated or non-iterated. The former separation relates to the degree of domain knowledge exploitation and dependency and the latter to incremental refinement. In the related literature, parallel multimodal fusion methods are commonly grouped according to at which point of analysis the fusion takes place, i.e., the level of the fusion [4–9, 20, 27].

Multimodal fusion can be applied in different stages of the content analysis chain. The choice of the stage or level of fusion affects among other things the effectiveness in exploiting the relations and dependencies between the modalities. Raw sensor data retains all information content of the modalities [9], but the inefficiency of the original representations for many inference tasks as well as the large share of irrelevant information along with the challenges of asynchrony and incommensurability usually makes this level unsuitable for direct fusion in any higher-level knowledge extraction tasks. Going up in the abstraction levels enables the refinement of the data representations to alleviate the aforementioned problems and to reveal modality relations. On the other hand, the refinement of the data representations can also accidentally remove some relevant underlying dependencies between modalities.

Different fusion level categorizations have been presented in the literature. Most common categorization is between applying the information fusion before decision making, and combining decisions of individual modalities [5–9, 20, 27]. Instances of the former have been termed feature level fusion, early fusion, early integration, feature integration, direct identification, pre-mapping or -classification fusion, or data to decision fusion, whereas the latter approach is known in the literature as late fusion, late integration, classifier fusion, decision (level) fusion, separated identification, or post-mapping or -classification fusion.

Some works also mention so called intermediate or classifier level fusion, where specific modeling approaches jointly combine and classify unimodal features [6] [8]. Distinction is made in [4] between weak fusion having separate likelihoods and processing chains for the modalities and strong fusion, where the joint likelihood of the modalities is non-separable and has a single prior. In weak fusion the fusion is done after obtaining the unimodal estimates. The weak fusion corresponds roughly to late fusion, and strong fusion to early fusion. The authors also mention an intermediate case, where the likelihood factors into two modality specific terms. Additionally, most categorizations point out the use of hierarchical hybrid combinations of the different fusion levels [5–8]. In finer categorization early fusion has been divided into feature-level fusion as well as signal enhancement and sensor level fusion (i.e., raw sensor data fusion), where information is combined at the raw sensor level prior to feature extraction [6, 9]. Shivappa *et al.* [6] additionally consider semantic level fusion, where high-level semantic interpretation of the decisions of different modalities is combined. Figure 1.2 presents the categorization of fusion levels.

The fusion of raw sensor data prior to any feature extraction is not very usable for higher-level inference, but can be used as a preprocessing component for hierarchical or hybrid systems, such as video-aided beamforming or visual target tracking with the

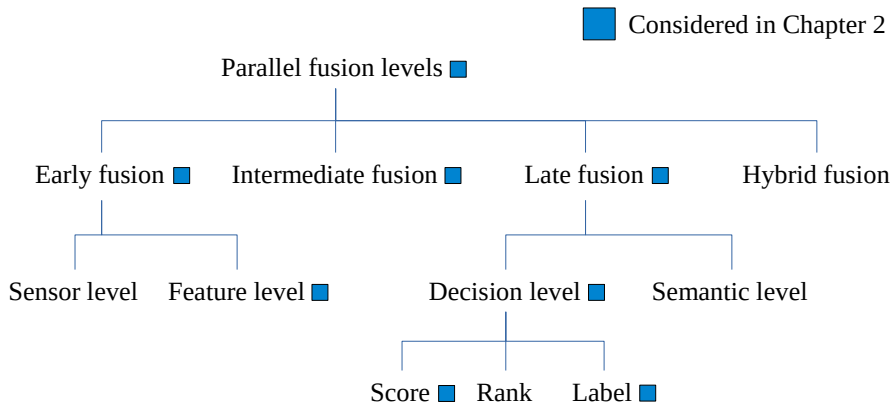

Figure 1.2: Hierarchical categorization of fusion levels found in the literature.

help of audio localization [6]. Sensor level fusion is rarely done between considerably different modalities – typically either multiple identical sensors or multiple consecutive measurements with a single sensor are used [9].

In feature level fusion the data of each modality is refined to a more suitable representation for a certain task and the chosen modeling approach – yet the multimodal information fusion happens before any explicit decision making. One of the simplest and most commonly used approaches to feature level fusion is concatenating the representations of the modalities and feeding this stacked representation to a decision making unit for fusion [6]. However, simple concatenation makes it burdensome to reveal semantically relevant but nonlinear relations between the modalities [21]. More elaborate approaches combine the modality representations by various normalization, transformation, and reduction schemes [9]. Fusion before the decision making means that only a single modeling process is required, which can be considerably faster than training separate decision logic for each modality [28]. However, the possibly higher-dimensional multimodal representation might slow down the fuser training if the chosen algorithm has scalability issues regarding the dimensionality. The increased dimensionality may also lead to overfitting, as generally more data is needed for satisfactory modeling with increased dimensionality. This issue can be alleviated with feature selection methods or other dimensionality reduction techniques, such as principal component analysis (PCA), which is a technique for finding a linear projection to map the data to a lower-dimensional space that retains the dominant variations. Feature level fusion is advantageous if the representations of the modalities along with the chosen fusion model allow efficient discovery of dependencies and covariations between the modalities [6]. One of the main drawbacks is that the different modalities need to be transformed into compatible form and synchronized, which might be laborous and require some impairing compromises. The form unification and dependency discovery is likely to become increasingly complicated with larger amounts of modalities [5]. Incorporating dynamic adaption between the modalities may also limit the choice of the fusion algorithm [6]. Namely, the methods need to allow the weighting of subparts corresponding to specific modalities in the joint representation,

which might be challenging or even impossible if the modalities are transformed into a distributed representation with no clear modality separation. Generally, feature level fusion also lacks in modularity, as modifications – e.g. the addition of a new modality – may require retraining the whole decision logic [4].

Decision level fusion processes the different modalities separately up to the point, where decisions are made from the refined data of each modality. The decisions are then combined with a fusion procedure. Decision fusion allows flexibility in customizing the decision logic to best suit each type of modality. Modality reliability adaption is also relatively easy as the separate decisions can be weighted prior to or during the fusion. Different training phases are needed for each of the modalities, and training models for new modalities scales linearly in the number of added streams [6]. Yet, in some cases training separate models for the modalities can be more efficient than training a single model for a higher-dimensional multimodal representation – especially if the fusion of separate decisions is done with simple arithmetic rules that need no training phase. In this case, also the addition of new modalities incrementally is more straightforward as only the decision logic for the added modality needs to be trained, as opposed to incorporating the new modality into a multimodal representation, which would require retraining of the whole fusion system. Decision level fusion has no direct means for utilizing feature level correlations [5]. The degree of relevant correlations and thus their value depends on the task and the modalities, but regardless this information is lost during the decision making process.

The incompatibility of the fusion input is also much less of a problem on the decision level compared to the feature level. Diverse representations with different units, scales, orders, and rates are abstracted as decisions often by aggregating sequences of raw data, which alleviates the asynchrony issues by lowering the granularity [8]. However, in the context of classification different machine learning algorithms may produce their decisions at different levels of abstraction: in the form of soft scores for each alternative (e.g., class probabilities, likelihoods, or confidences), as a list of the alternatives ranked by their likelihood, or simply as the single most likely alternative [9, 26]. In [27] the terms decision fusion and opinion fusion are used to refer to the most likely alternative and soft score cases, respectively, as the former case outputs a single crisp decision whereas the latter gives an opinion with a degree of belief for all alternatives. Ranking can easily be derived from the soft scores and it is further trivial to pick the most likely decision from the ranked list. Traversing to the direction of lower abstraction level with higher information content is generally more challenging, but can be done at least approximately with some constraining assumptions, heuristics, or by estimation from example data [26]. Thus, combining decisions of different abstraction levels is most straightforward by converting the decisions of all modalities to the highest abstraction level among them.

Soft scores can be fused by countless methods including summing (or averaging), multiplying, maximizing, e.g., the maximum, median, or minimum score over the modalities, Bayes belief integration, fuzzy integrals and templates, using Dempster-Shafer (DS) theory, or by training any supervised learner for fusion [26, 29]. Many of the approaches either have implicit notion of weights or can be appended with weighting, e.g., by raising the modality scores to exponents defined by the weights [4]. Soft scores can be fused by countless methods including summing (or averaging), multiplying, maximizing, e.g., the maximum, median, or minimum score over the modalities, Bayes belief integration, fuzzy integrals and templates, using DS theory, or by training any supervised learner for fusion [26, 29]. Many of the approaches either have implicit notion of weights or can be appended

with weighting, e.g., by raising the modality scores to exponents defined by the weights [4]. Soft scores of different types – such as template match scores and probabilities – might additionally require normalization in order to balance the contribution of different modalities in the fusion. Ross *et al.* [30] term such fusion techniques as *transformation based score fusion* and distinguish them from *density-based score fusion*, where the fusion is done with a generative approach using the Bayesian decision rule, and *classifier based score fusion*, where a classifier is trained to conduct the fusion.

Ranked list fusion has the advantage over soft scores that no such normalization is required regardless of how the ranking has been acquired [9]. Ranked list fusion is typically done either by reducing the set of the alternative hypotheses based on component-wise rank thresholds defined from training data or by forming a reranked list from the lists of the different modalities and picking the top entry on this reranked list as the fusion output, or a combination of the two approaches [26, 27].

Single decisions can be fused, e.g., with various voting methods (majority voting, weighted majority voting, AND fusion, OR fusion), as well as with Bayesian decision fusion, DS theory of evidence, or so called behaviour knowledge space method [9, 26, 27]. The voting methods are based on historamming the component decisions possibly multiplied with weights. Many of the fusion algorithms for single decisions have means for refusing to output any fused decision in case of too low degree of consensus among the components.

In semantic level fusion information is fused after semantic interpretation of the content in the sensed signals of the modalities [6]. E.g., the recent work in image semantic labeling can be considered as fusion on the semantic level [31, 32]. Techniques that jointly perform the decision making and modality combination have been termed intermediate fusion. Their development has largely been motivated by trying to combine the advantages of early and late fusion. Intermediate fusion handles the input modalities separately, which allows increased flexibility in modality reliability adaption, while trying to find and utilize dependencies for optimal fused decisions. Certain methods can also implicitly account for some degree of asynchrony [8]. The flexibility comes with the price of specificity in the modeling requirements, which severely limits the amount of applicable modeling approaches [6]. Due to the complexity of the models they might also be difficult to train efficiently. Besides intermediate fusion, various hierarchical and hybrid fusion methods have been adopted to allow even greater flexibility in combining the advantages and suppressing the shortcomings of individual fusion schemes and levels.

In his recent survey Zheng [17] describes a stage-based strategy for cross-domain data fusion, which relates to sequential fusion. In the stage-based fusion, diverse datasets describing some aspect of a common domain are used sequentially for refined segmentation, finding retrieval cues, or inferring hidden knowledge about the domain. As an example, traffic anomalies can be detected and described by first detecting irregularities from vehicle GPS and road network data, and then trying to identify the anomaly by searching for social media content with the location of the anomaly and keywords such as *parade* or *disaster*. The author distinguishes stage-based fusion from feature level-based and meaning-based fusion, which roughly correspond to early and late fusion, respectively. Specifically, in this context feature level fusion means methods that treat the data simply as numbers and do not aim at interpreting or understanding the content – this task is left for the processing stages after the fusion. These methods, including concatenation, sparsity-regularized feature selection and combination, as well as multimodal hierarchical repsentation learning, are data-agnostic and thus highly generic, but in some tasks interpretation of the content and domain knowledge can be a valuable asset. Meaning-

based fusion methods try to explicitly tap into this higher-level information prior to the fusion. They are further grouped as multi-view-, similarity-, probabilistic dependency-, and transfer learning-based methods.

Multi-view-based fusion uses multi-view learning techniques such as cotraining. Similarity-based fusion models the underlying dependencies and correlations between different objects with methods like coupled collaborative filtering or manifold alignment – i.e., the multi-view extensions of collaborative filtering and manifold learning, respectively – to fuse multiple data sets describing them. In probabilistic dependency-based methods the fusion is done by modeling the probabilistic dependencies (rather than object similarities) between data sets using probabilistic graphical networks, such as Bayesian networks or Markov Random Fields. Transfer learning-based fusion transfers knowledge gained from one data set, distribution, domain, or task to another, with various techniques available depending on the differing aspects of the source and target task – such as missing modalities – as well as on the availability of labeled data.

No single fusion level or method has been shown to consistently outperform the others in wide selection of tasks, modalities, and data sets. The optimal choice of fusion level as well as the overall fusion architecture depends on the application and the information sources available. In practice, often the fusion strategy is chosen mainly according to the chosen modeling framework, which has been picked based on the task [6].

1.2.4 Fusion models

Bezdek *et al.* [33] divide decision fusion methods to three categories by their degree of adjustment to the data and decisions of the modalities. The first category consists of methods using a fixed non-trainable operator. In the second category the fuser is trained separately from the modality decision making algorithms. In the third category the modality decision logic and the fusion is optimized jointly from the training data, which is related to intermediate fusion approaches. Atrey *et al.* [5] use a problem space categorization to distinguish between rule-based, estimation-based, and classification-based approaches.

Rule-based fusion

Rule-based fusion uses simple fixed rules, such as arithmetic or logical operators, or order statistics. Thus, usually there is no need for training any decision logic from example data. However, e.g., the use of modality-specific weights often requires optimization. One widely used rule-based method is linear weighted fusion, where the modalities are weighted and summed together. It is simple both in terms of implementation and computational burden, applicable to late as well as early fusion with proper synchronization and representation matching of the modalities, and inherently accounts for modality adaption [5]. However, the adaption by weighting requires specifying the weights, which is often nontrivial. Weighted average and sum fusion are special cases of linear weighted fusion. Examples of other common rules include product, maximum, minimum, logical AND as well as OR operators, median, majority voting, and domain-specific custom-defined rules. Custom rules are typically easily understandable, flexible to add, and effective under the domain assumptions, but require a domain expert to define, might fail if the domain assumptions are not met or conditions change, and are seldom generalizable outside the specific domain [5, 12].

Classification-based fusion

In classification-based fusion the fusion problem is formulated as a (usually supervised) classification problem, and a classifier is trained from example data to perform the fusion. The classifier can be trained for early fusion, e.g., by concatenating the representations of the modalities and feeding the concatenated representations and the corresponding class labels to the classifier. If it is inconvenient to get the modalities synchronized and in similar enough representation for feature level fusion, separate decision logic can be applied to all modalities, and their decisions used as input to a fusion classifier. This technique is known as stacking and the fusion classifier sometimes called a meta-level classifier.

SVM has been a common choice as a fusion classifier for multimodal multimedia tasks [5]. Other commonly used approaches include various types of artificial neural networks (ANNs), Bayesian inference, as well as DS theory. Temporal or otherwise sequential data fusion is often done with dynamic Bayesian networks (DBNs), which are a subset of sequential directed probabilistic graphical models. HMM is a simple type of DBN that has been widely applied for various multimedia sequence analysis tasks including multimodal fusion.

Data-driven hierarchical representation learning has lately been applied for multimodal tasks [14, 21, 31, 34–37]. Ngiam *et al.* [14] experiment with deep autoencoders on the task of audio-visual speech recognition in various multimodal learning settings, namely multimodal fusion, cross-modality learning, and shared representation learning. Frome *et al.* [31] map visual representations learnt with a convolutional neural network (CNN) into an embedding vector space learnt from skip-gram text modeling for extending the classification capacity of the CNN trained for 1000 classes to 20000 classes by similarity-based retrieval in the embedding vector space. Srivastava and Salakhutdinov [21] use a probabilistic generative model called deep Boltzmann machine (extending restricted Boltzmann machine (RBM) by stacking) for learning fused representations between images and text, and experiment with predicting missing modalities. The minimization of the variation of information measure is proposed in [34] as training objective for multimodal fusion with the aim to improve inference with missing modalities. Deep canonically correlated autoencoders are proposed for cross-modality learning in [35]. Wang *et al.* [38] propose unsupervised and supervised methods for learning latent representations for cross-modal retrieval between text and images. The unsupervised approach uses stacked autoencoders, as opposed to a CNN and a neural language model used in the supervised case. After training separate unimodal models, the learning of a latent multimodal representation is guided by a loss function term that tries to map semantically similar unimodal data close to each other in the latent space. Feng *et al.* [36] propose to add a correlation constraint between the representations trained for two modalities with RBMs. This allows the coupling of the representation learning at every layer, when stacking multiple RBMs, as opposed to the approach of learning a joint representation after independent unimodal representation learning. In [37] separate stacked contractive autoencoders are used for representation learning from video frames, audio, and text. The unimodal representations are then concatenated for learning a multimodal contractive autoencoder. Although the so called deep learning models have proven powerful and effective for modeling highly nonlinear dependencies between modalities and pushed the state-of-the-art in many fields, they usually require large data sets and considerable computation power to properly optimize the numerous parameters of the models. The model complexity may also prevent their utility in environments with limited memory

or computation power, or in applications with hard real-time constraints. Additionally, the black box nature of the methods along with the difficulty to interpret the internal representations might be problematic in some domains or use cases.

Cotraining is a technique from multiview learning, where classifiers are trained on two views from a small labeled data set, and then iteratively improved on an unlabeled data set by alternating between classifying the samples that the view is most confident about, and retraining from the dataset appended with the new samples pseudolabeled by the other view [8]. Another common multiview learning approach called co-regularization includes a view disagreement penalty term to the objective function to be minimized during training [39]. In [17] multiple kernel learning (MKL) and subspace learning are mentioned as common multiview learning approaches besides cotraining methods. MKL can be used to extend SVM to multiview or multicue setting by allowing the use of different kernels for different data sources and finding their optimal linear or nonlinear combination. MKL can be considered an intermediate fusion method as the different sources are modeled separately and fused within the same decision unit. In the multiview context, subspace learning comprises of multiview representation refinement by cross-correlation finding methods such as canonical correlation analysis (CCA) [40] or its nonlinear extensions by kernels [41] and neural networks [42]. The methods transform the unimodal representations to a common feature space enabling cross-modal retrieval and comparison.

Estimation-based fusion

Estimation-based fusion is typically used in tasks, where the state of a system needs to be estimated from noisy measurements from multiple modalities. A common such task is tracking a moving target with various sensors. Bayesian filtering techniques can be considered as the *de facto* standard for such tasks [43]. These include the Kalman filter (KF), which produces optimal estimates for linear systems with additive Gaussian noise within the process and in the observations. In case of nonlinear systems, nonlinear extensions of KF, such as extended Kalman filter (EKF) and unscented Kalman filter (UKF) are commonly used. In EKF the Jacobian of the nonlinear mapping with respect to the system state is used to linearize the system around the most recent estimate, so the nonlinearities need to be differentiable. UKF uses a sampling technique to pick a set of points around the current estimate, propagate them through the nonlinearity, and calculate their mean and covariance as approximations for the mean and covariance of the unknown target distribution transformed by the nonlinearity. UKF is applicable to non-differentiable nonlinearities and does not require the calculation of the Jacobian, which might be laborious or even impossible for many practical problems. EKF is used in [44] for object tracking with a camera and a laser range scanner. The modalities are fused by concatenating the unimodal EKF measurement model matrices as a multimodal measurement model.

In non-Gaussian systems particle filter (PF) is a commonly used tracking technique. PF estimates the system state using a set of particles that are propagated according to a dynamic model, weighted according to their likelihood of accounting for the observations, and resampled according to the weights. The PF estimate is obtained as the weighted average state of the particles. It approaches the Bayesian optimal estimate for nonlinear non-Gaussian processes with sufficient amount of particles [5]. In [45] object position likelihoods from stereo cameras and audio beamforming are combined by multiplication in a PF setting. Nickel *et al.* [46] combine visual person detection and sound source localization

for PF multi-person tracking with a setup of four cameras and three microphone arrays. The particles weights are assigned as a weighted sum of the likelihoods of the different sensors. The modalities are weighted based on the average confidence of the sound source localization over all microphone pairs. In [47] person tracking is done with a PF from video and RFID sensing. The particle likelihoods are estimated as a weighted sum of the unimodal likelihoods. In the audiovisual PF-based tracking system of [48] the fusion is done by multiplying the unimodal particle weights, but additionally the sound direction of arrival estimates are explicitly used to modify the particle propagation.

1.3 Objectives of the thesis

The main objectives of the doctoral research work presented in this dissertation are

1. to examine the gains of utilizing multimodal processing for video analysis tasks,
2. to study the interaction and correlations between different modalities,
3. to study the complementarity, diversity, and reliability of different information sources for various multimodal video analysis tasks under varying conditions,
4. to examine the feasibility of data-driven modeling of music-based cues and music meter based temporal granularity for timing shot cuts to unedited concert video material,
5. to investigate guidelines for choosing the optimal information fusion approach to a given multimodal visual analysis problem with a given set of modalities, and
6. to develop efficient implementations for multimodal video analysis algorithms on specific real-world problems.

1.4 Outline of the thesis

Chapter 2 presents methods for multimodal fusion in selected audiovisual classification tasks. In the first task, various rule-based audiovisual fusion methods are compared to classifier-based fusion for environment classification in mobile videos. In the second task, the sport type is estimated from sets of user generated multi-camera videos and the corresponding recording device motion signals. In chapter 3 a framework is presented for modeling the timing of view switches with regard to music rhythm in professional concert videos, and for synthesizing cut times for sets of unedited multi-camera videos recorded by concert-goers. Conclusions and possible directions for future work are discussed in chapter 4.

1.5 Main results of the thesis

The multimodal fusion work contributed to the field of multimodal analysis by comparing a comprehensive set of fusion and modality adaption techniques, highlighting feasible multimodal fusion techniques for the considered problems, and verifying the advantages of multimodality on two original data sets. In both the environment and sport type video classification tasks, training a dedicated classifier for fusing the modalities (either from concatenated features or unimodal likelihoods) seemed to work better than rule-based

fusion – even with weight optimization from data. Yet, for aggregating single-video sport type predictions to a collective prediction for an event, where a set of videos have been recorded, simple majority fusion methods operating on crisp predictions gave the overall best performance over methods working with likelihoods.

User studies conducted for the concert video cut timing work validated the feasibility of the example-based modeling and the analysis granularity based on music meter estimation. The users clearly preferred the proposed approach over a baseline method, and cuts timed from the proposed approach were evaluated more pleasant than manually assigned cuts in nearly half of such comparisons. The proposed framework is a considerable alternative to the commonly used absolute time based cut timing in automatic concert video editing. It can also be adapted to other music domains, such as non-live music videos or dance performances, by training on data from the given domain.

1.6 Author's contributions to the publications

In publication 1 the author was responsible for visual feature extraction using the MUVIS system [49], the overall fusion system design and implementation, as well as the experiments. The audio likelihoods were provided by the system presented in [50].

In publications 2 and 3 the author collaborated with Francesco Cricri on the planning and implementation of the various considered parallel fusion approaches and the quality-based dynamic weighting. Francesco Cricri was in charge of extracting the unimodal features and the modality qualities as well as for the sequential fusion experiment. The audio feature extraction and quality estimation was courtesy of Jussi Leppänen. Stefan Uhlmann aided with the intermediate fusion framework used in publication 3.

In publication 4 the author was responsible for the experiments, overall system design and implementation, except for the music meter analysis, which was implemented by Antti Eronen, and the audio change point and music section detectors, which were implemented by Jussi Leppänen. The overall example-based modeling idea was formed and refined in discussions with Igor D. D. Curcio and Antti Eronen and the starting point basic Markov chain cut pattern modeling idea came from Antti Eronen.

2 Multimodal fusion for video classification

This chapter presents the methodology used on two multimodal video classification tasks. In the first task described in more detail in publication 1, audio and video are combined for environment classification using rule-based fusion with static modality adaption as well as stacked SVM fusion. The second task presented in publications 2 and 3 consists of classifying the sport type of a set of videos from a common sport event based on video, audio, and recording device motion sensors. A large set of rule- and classification-based fusion strategies is compared in the second task.

2.1 Methods

In this section a set of tools used in the fusion systems is briefly described. The tools include GA optimization as well as SVM classifiers.

2.1.1 Genetic algorithms

GAs are a class of search and optimization algorithms from the field of evolutionary computation [18]. A simple genetic algorithm consists of presenting solutions to an optimization problem as binary vectors, which are evolved by selection, cross-over, and mutation operators. The selection operator simply selects a subset of the current solutions to move on to the next iteration according to some fitness criterion. If the original optimization problem is fast enough to evaluate, it can be used as the fitness criterion – otherwise a simpler approximating criterion could be used instead. The selection operation steers the optimization towards the most promising solutions at each iteration. Cross-over splits the binary vectors of a pair of solutions at the same elements and switches the remaining subvector of one solution with the corresponding part of the other solution. This is inspired by the cross-over of genes in natural reproduction. Cross-over introduces new solution candidates as combinations of old solutions. Mutation operation flips a bit in a solution vector according to a small probability. Mutation provides means for exploring the search space in regions otherwise unreachable or escaping local fitness maxima (or error minima).

2.1.2 Support Vector Machines

Support vector machine (SVM) is a discriminative, supervised machine learning technique that tries to reduce the generalization error by using decision boundaries that maximize the separation between the classes in the training data [51]. This is achieved in binary linearly separable problems by setting the decision boundary so that the margin between

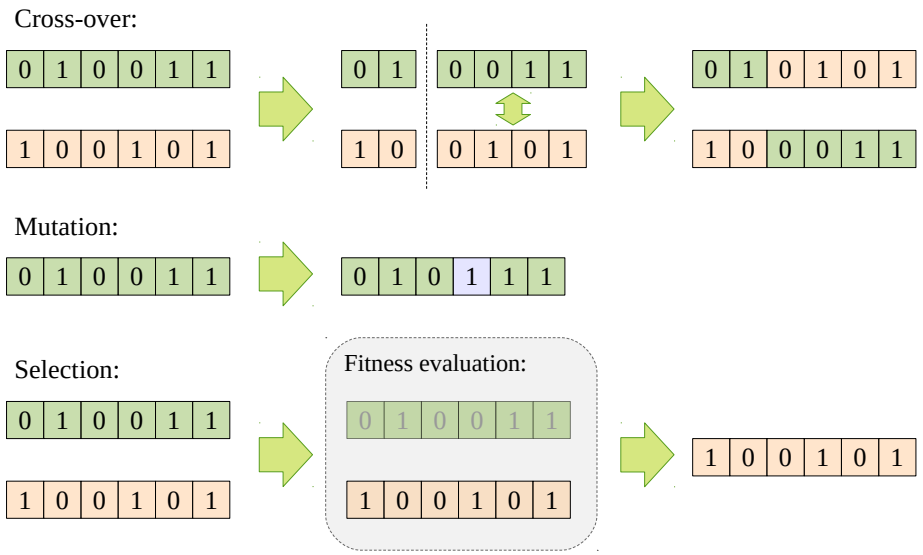

Figure 2.1: Basic operations of GAs are cross-over, mutation, and selection.

the boundary and the closest training examples from each class is as large as possible. The training data points lying on the margin edges are called support vectors.

Given a data set of N features $\mathbf{x}_n, n = 1, ..., N$ belonging to two classes as indicated by $t_n \in \{1, -1\}$ SVM is trained by maximizing equation

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{2.1}$$

with respect to the constraints

$$a_n \geq 0, \quad n = 1, ..., N, \tag{2.2}$$

$$\sum_{n=1}^{N} a_n t_n = 0. \tag{2.3}$$

The parameters a_n are so called Lagrange multipliers commonly used for reformulating problems of finding extrema in constrained multi-variable problems and $k(\mathbf{x}_n, \mathbf{x}_m)$ is the kernel function defined as the inner product $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^{\mathsf{T}} \phi(\mathbf{x}')$. The fixed nonlinear feature space mapping $\phi(\cdot)$ defines a given kernel. For linear SVM the $\phi(\cdot)$ is simply the identity function so the kernel becomes the inner product $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\mathsf{T}} \mathbf{x}'$. The SVM training problem is in the form of a quadratic programming problem. Many algorithms with different memory requirements and computational complexity exist for solving such

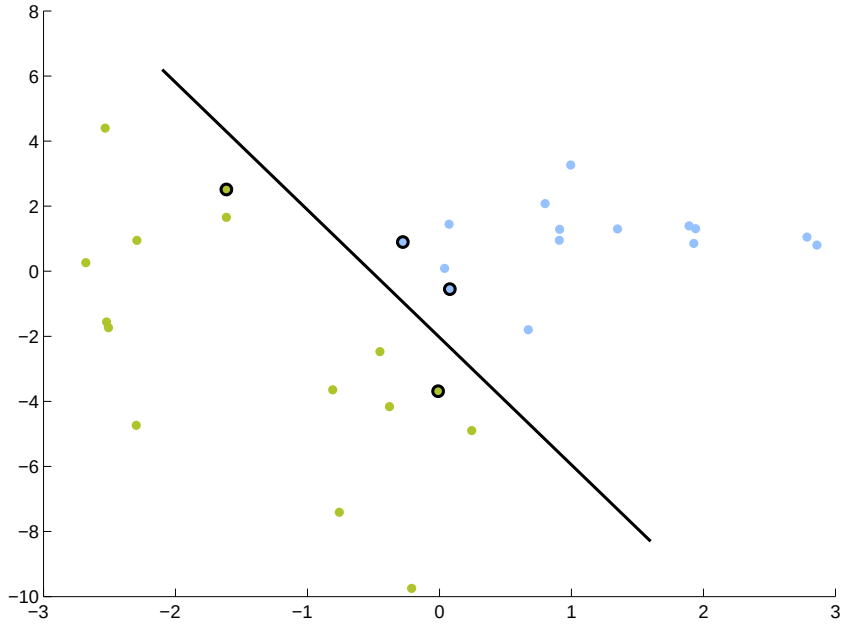

Figure 2.2: The SVM basic principle exemplified in a linearly separable binary toy problem. The support vectors are shown encircled.

problems. One popular algorithm for SVM training is sequential minimal optimization [52].

Prediction of a sample $\mathbf{x}$ is done by evaluating the sign of equation (2.4).

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \tag{2.4}$$

Only the support vectors have a_n values greater than zero, which greatly reduces the amount of terms in the sum of equation (2.4).

Nonlinear decision boundaries can be achieved with the use of nonlinear kernel functions, i.e., choosing $\phi(\cdot)$ of a specific nonlinear form. The kernels effectively map the input data into higher-dimensional kernel space, where many nonlinear problems can be made linearly separable. The linear decision boundary in the kernel space results as a non-linear boundary in the original feature space. The formulation of the kernels as the inner product of two mapped features enables the implicit use of the high-dimensional kernel space without explicitly calculating the representations of the features in this space. This is known as the kernel trick. In fact, some kernel spaces even have infinite dimensionality - yet they can be efficiently used via the kernel formulation. Besides the linear kernel, commonly used kernels are, e.g., polynomial kernel of degree M

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\mathsf{T}\mathbf{x}' + c)^M, \tag{2.5}$$

the sigmoidal kernel

$$k(\mathbf{x}, \mathbf{x}') = \tanh(a\mathbf{x}^\mathsf{T}\mathbf{x}' + b), \tag{2.6}$$

the exponential kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\theta|\mathbf{x} - \mathbf{x}'|), \tag{2.7}$$

or radial basis function (RBF) kernels, such as Gaussian kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right). \tag{2.8}$$

For problems with considerably overlapping class distributions, perfect separation of the training data might be difficult to achieve even with the nonlinear kernel mapping. Perfect separation in such a case would result in severe overfitting of the training data set. For such cases it is possible to add a cost term that penalizes samples that reside on the wrong side of the decision boundary. The cost depends on the distance of such samples to the decision boundary. This creates sort of a soft margin, where samples are allowed to reside at the wrong side of the decision boundary but this is penalized, as opposed to the hard margin of the nonoverlapping case. Soft-margin SVMs are also trained with equation (2.1), but with slight modification to the constraints, namely equation (2.2) is replaced by

$$0 \leq a_n \leq C, \quad n = 1, ..., N, \tag{2.9}$$

where C is a tradeoff parameter for balancing between increasing the margin and penalizing the misclassifications in the training data. Prediction is again done by examining the sign of equation (2.4). It is also possible to use two different tradeoff parameters C^+ and C^- for training samples with targets $t_n = 1$ and $t_n = -1$, respectively. This can be advantageous if the training data is imbalanced, i.e., contains different amounts of examples from the two classes, or if misclassification of one class would have more severe consequences than the other.

SVM outputs only a fixed decision and no estimate for the confidence of the prediction, such as a posterior probability [10]. However, methods have been proposed for estimating posterior probabilities based on decisions on a validation data set or cross-validated decisions from the training data [53, 54].

Extending SVM classification for more than two classes is typically done by training multiple two-class classifiers and aggregating their outputs. In one-against-all strategy a single pair-wise classifier is trained between the data of one class and all the remaining data. This strategy has the issue that the training problem becomes increasingly imbalanced with increased amount of classes. This can be compensated with adjusting the values of C^+ and C^- or by various methods for oversampling the minority and/or undersampling the majority class (i.e., the temporary class formed from all remaining data). One-against-one strategy trains pair-wise classifiers for all pairs of classes and counts the decisions for each class. Compared to one-against-all, considerably more classifiers need to be optimized and evaluated during training and testing. However, the problems are generally simpler and

more balanced. In both multi-class strategies the aggregation of the outputs of individual classifiers might indicate a tie between multiple classes. In one-against-all strategy ties occur if more than one classifier outputs the class that it was trained to separate from the remaining data. In one-against-one strategy a tie occurs if multiple classes get the same amount of votes. One option to resolve the ties is to estimate the posterior probability or some other confidence value for the decisions, and pick the class with the highest confidence.

2.2 Audiovisual video context recognition

The automatic recognition of the usage context, i.e., the location, environment type, momentary conditions, etc., has many uses for smart devices and agents. This information can be complementary to the location information from positioning systems, such as GPS or any indoor positioning systems. As an example, GPS can localize a user to a certain stadium, whereas audiovisual information can reveal whether there is a concert or a sport event taking place at the stadium. This chapter presents methods from publication 1 for offline audiovisual environment classification from mobile videos. Specifically, a system is described for classifying between 21 everyday contexts from video and the corresponding audio track. Various late fusion techniques are compared for combining the information from the audio environment classifier and multiple visual context detectors.

2.2.1 Unimodal descriptors

The visual modality is described with various global visual features of the video keyframes. The features include color histograms in HSV, RGB, and YUV color spaces, gray-level co-occurence matrix (GLCM) [55], ordinal co-occurrence matrix (ORDC) [56], as well as MPEG-7 edge histogram [57]. A SVM classifier with a RBF kernel is trained separately for each visual feature type for distinguishing between the context classes. The auditory scene analysis is done with the audio-event detection based context recognition system proposed by Heittola *et al.* in [50]. The system classifies between everyday audio environments by recognizing audio events, forming their histogram, and comparing the histograms with the cosine distance to environment template histograms calculated from a training data set. The classification is done using k-NN based on the cosine distances. The SVMs are trained on each visual feature as well as the audio-based recognizer output likelihoods for each of the context classes calculated from the classification confidences.

2.2.2 Audiovisual fusion

Simple rule-based fusion schemes are compared to classification-based fusion with a SVM. The considered rule-based methods include majority voting, sum of likelihoods, product of likelihoods, as well as maximizing the minimum and maximum likelihood. The individual modalities are weighted prior to applying the fusion rules. The weighting is done in a static manner by finding modality specific weights offline with GA optimization, which was chosen due to the multimodal nature of the problem resulting likely in a multimodal distribution of the solution space. The weights of the different modalities are concatenated and used as the genome in the GA optimization carried out with the GAlib library [58], which internally handles the encoding and decoding between the real-valued weights and the binary valued genomes. The fusion performance on a holdout weighting data partition is used as the fitness function for the GA search. The weights are optimized separately for each fusion rule, as the weighting is done differently for different rules. In majority

voting a vote for class j from modality i increments the accumulated weight of the class by the motality weight w_i. The class with the highest accumulated weight is then chosen as the fused output. Given the likelihood l_{ij} of class j from modality i, the corresponding weight w_i, and maximum weight value a, the weighted likelihoods $\tilde{l}_{ij}$ are obtained in sum of likelihoods, product of likelihoods, maximum of minimum likelihood, and maximum of maximum likelihood fusion by equations (2.10), (2.11), (2.12), (2.13), respectively.

$$\tilde{l}_{ij} = w_i l_{ij} \quad 0 \leq w_i \leq a \tag{2.10}$$

$$\tilde{l}_{ij} = l_{ij}^{w_i} \quad 0 \leq w_i \leq a \tag{2.11}$$

$$\tilde{l}_{ij} = \begin{cases} (1 - w_i)\mu_j(i) + w_i l_{ij} & 0 \leq w_i \leq 1 \\ l_{ij}^{w_i} & 1 < w_i \leq a \end{cases} \tag{2.12}$$

$$\tilde{l}_{ij} = \begin{cases} (1 - w_i)\mu_j(i) + w_i l_{ij} & 0 \leq w_i \leq 1 \\ w_i l_{ij} & 1 < w_i \leq a \end{cases} \tag{2.13}$$

In equations (2.12) and (2.13) $\mu_j(i)$ is the average of the likelihoods of all modalities for class j excluding modality i. This is formally defined as:

$$\mu_j(i) = \frac{1}{N-1} \sum_{k=1}^{N} (1 - \mathbf{1}(i,k)) l_{kj}, \tag{2.14}$$

where N is the amount of modalities and $\mathbf{1}(\cdot,\cdot)$ is the indicator function returning one if the arguments are equal and zero otherwise. In sum of likelihoods fusion the weight scales the likelihoods of the corresponding modality affecting linearly the contribution of the modality in the fusion. In product of likelihoods fusion raising the likelihood to the power of the weight essentially affects, how many times the likelihood is considered in the product. Weights approaching zero make the weighted likelihoods approach unity for all classes, which reduces the contribution of the corresponding modality. A similar weighting approach was proposed in [59] for fusing acoustic and visual likelihoods in audiovisual speech recognition. In both maximum of minimum and maximum of maximum fusion schemes with weight values below one, smaller weight values push the weighted likelihood of class j from modality i towards $\mu_j(i)$, the modality-wise average likelihood excluding modality i. It thus becomes less likely for the weighted likelihood to be the smallest or largest among the modalities. In maximum of minimum fusion with weights greater than one, the likelihood gets raised to the power of the weight, which scales down the likelihoods of the modality, but does not change their ranking. Hence, it is more likely that over all modalities the smallest likelihoods come from this modality, yet the most likely class prediction from the modality does not change from the unweighted case. In maximum of maximum fusion, the likelihood gets multiplied by the weights, when they are larger than one. The larger the weight w_i, the more likely that the highest weighted likelihood over all modalities and classes comes from the modality i, and again the weighting does not change the likelihood ranking of the classes for a given modality.

In the alternative classification-based fusion scheme, the likelihoods of all modalities for all classes are concatenated and a SVM with RBF kernel trained to conduct the fusion.

Table 2.1: Correct classification rates of the individual experts. Reproduced with permission from publication 1.

EHD7	GLCM	HSV	ORDC	RGB	YUV	Audio
0.418	0.325	0.609	0.644	**0.653**	0.620	0.564

No explicit weighting of the likelihoods is used in this setting, as it is left for the SVM to find the optimal combination of modalities. This scheme thus avoids the GA optimization, yet requires training of the classifier.

2.2.3 Evaluation

The environment classification was evaluated on a set of 193 videos recorded in 21 everyday environments with a wearable camera and a mobile phone as well as a separate portable audio recorder. Visual feature extraction was done for 10161 video key frames, half of which were used for training and half for testing. The audio-based environment likelihoods were provided around the test key frames by the original authors of the context recognizer. In case the audio data was not available around the key frame (due to differences in the exact starting times and durations of the recordings of the two modalities), uniform likelihoods were used for the audio modality. The whole multimodal environment classification procedure was iterated 10 times with different random splits of the data and all reported results calculated as the average over the different runs. Table 2.1 shows the performance of the separate fusion components, i.e., the audio context recognizer and SVM classifiers trained on the different visual features. Table 2.2 shows the results for the fusion of the components with different amount of visual experts manually included in the order of decreasing individual performance. The last row shows the performance of the rule-based fusion methods using the average of optimal weights over the iterations. The fusers were optimized on half of the samples of the set used to evaluate the individual components, and the fused performance evaluated on the remaining half. More extensive results can be found in publication 1.

2.3 Multimodal sport type classification from video

Automatic sport video analysis for content structuring, indexing, and understanding has many uses for various interest groups. Athletes can quickly gain insight from their own recorded performances, coaches or other instructors can amass statitics on advantageous or avoidable techniques from large amounts of data, and sport enthusiasts may enjoy, e.g., automatic personalized summaries or informative visualizations appended on top of the video content. Content-based indexing also enables effective content retrieval. Domain knowledge about the rules and the typical course of events in a sport can be highly valuable for further analysis tasks due to the regulated nature of sports and often notable variation of the regulations between sport types [60]. As an example, the rules and the overall goal of sports in athletics differ greatly from those of ball games, and furthermore there is considerable variation within the categories. The domain knowledge can be obtained by identifying the sport type from the content. Although, for larger events the sport type can be retrieved by metadata such as time and location, this information might not reveal the sport type reliably enough for smaller events or if diverse sports

Table 2.2: Correct classification rates of the different fusion methods. Reproduced with permission from publication 1.

Experts	Maj. voting	Sum	Product	Min.	Max.	SVM
6 vis. + aud.	0.772	0.775	0.761	0.737	0.714	0.833
6 vis.	0.711	0.743	0.758	0.739	0.706	0.785
5 vis. + aud.	0.772	0.816	0.761	0.739	0.716	0.837
5 vis.	0.711	0.743	0.757	0.739	0.706	0.781
4 vis. + aud.	0.771	0.824	0.773	0.736	0.715	**0.844**
4 vis.	0.713	0.743	0.757	0.738	0.706	0.778
3 vis. + aud.	0.756	0.836	0.785	0.734	0.711	0.841
3 vis.	0.686	0.731	0.746	0.734	0.700	0.763
2 vis. + aud.	0.720	0.830	**0.835**	0.740	0.705	**0.844**
2 vis.	0.652	0.721	0.738	0.726	0.695	0.752
1 vis. + aud.	0.652	0.784	0.801	0.728	0.734	0.828
Optim.	**0.774**	**0.837**	0.817	**0.752**	**0.771**	–

take place in a common location. Unedited mobile video recordings contain no editing patterns typical to broadcasts of a certain type of sports or augmented information, such as game scores or lap times.

This chapter presents multimodal sport type classification of events concurrently captured on video by multiple users. The task is, for an event E_l consisting of a set of N_i video clips[1] $E_l = \{v_{li}\}, 1 \leq i \leq N_i$ from the lth event, predict the sport type of the event using video and audio information as well as recording device tri-axial magnetometer and accelerometer sensors. The magnetometer reports the device horizontal orientation with regard to the North Magnetic Pole and accelerometer the acceleration in the three spatial dimensions perpendicular to each other in the coordinate system of the device. The static gravitational acceleration of 1g (i.e., 9.81m/s^2 at sea level) can be used to retrieve the device orientation. The sensor signals are regarded as a single camera motion modality by concatenating the various features extracted from them to a joint sensor modality representation. Thus the fusion is conducted between audio, video, and sensors in publication 2. However, in publication 3 global spatial and local spatio-temporal features are treated as two different modalities resulting in the fusion of four modalities. Odd number of modalities can be advantageous for majority voting fusion, as it is less likely for more than one class to get the highest amount of votes. A comparative evaluation of the approaches has been carried out on a data set collected using mobile phones containing a tailored video recording software for capturing the sensor signals along with video data. The data set consists of videos from six different sport types, namely, soccer, American football, basketball, tennis, ice hockey, and volleyball. It is assumed that only a single sport type takes place at a certain recording location and time, which is a reasonable assumption for the considered sport types (as opposed, e.g., to track and field events). This simplifies the problem in three ways: First, each individual video clip v_{li} can be assumed to contain at most one of the considered sports (although they may also contain non-sport segments considered as noise) and thus the videos do not need to be segmented,

[1] In the remainder of this chapter, a single recorded video along with its audio track and the corresponding sensor signals is collectively meant by the term *video clip*, unless explicitly stated otherwise.

but can be classified as a whole. Second, a single prediction can be aggregated for each event E_l from the predictions of the videos v_{li}. Lastly, the mapping of individual videos to the events can be obtained from the recording time and location, and does not need to be estimated, e.g., by content-based clustering. This chapter concentrates mainly on the multimodal fusion and video-to-event aggregation aspects of publications 2 and 3. For additional details on other parts of the work – besides taking a look at the publications – the interested reader is adviced to refer to the dissertation of the first author of the publications [61].

2.3.1 Modality representations

In publication 2 the data from the different modalities is represented in the following form. Frames from the video clips are sampled with the sampling period of one minute. The obtained frames are represented by concatenating the following MPEG-7 global visual descriptors [57]: dominant color, color layout, color structure, scalable color, edge histogram, homogeneous texture. The audio tracks are described with 12 Mel-frequency cepstral coefficients (MFCCs) calculated from consecutive monophonic 40 ms audio frames without overlap. The magnetometer and accelerometer signals are obtained with 100 ms and 25 ms sampling periods, respectively. Various features describing panning movements of the recording device, i.e., rotating the device around the axis aligned with the direction of gravity, are extracted from the magnetometer signal of each video clip. These include panning rate; average, median, minimum, and maximum panning extent, panning speed, and panning duration; panning direction change rate; discrete cosine transform (DCT) components; and magnetometer variance. Similarly, the accelerometer signal corresponding to a given video clip is used for describing the camera tilting, i.e., rotations resulting in upwards or downwards motion in the video, by tilting rate; average, median, minimum, and maximum tilting speed and tilting duration; DCT components; and accelerometer variance. The resulting features of both sensors are concatenated as the camera motion feature of the clip. The motivation behind the camera motion features is to investigate possible discriminative patterns in typical camera motions in different sports due to people orienting the cameras to follow the game flow.

In publication 3 the global visual features are complemented with local spatio-temporal features space-time interest points (STIP) [62]. The sensor features are used in a sequential fusion manner to identify the segments of the video clips with stable visual content, and the STIP features are extracted only from such segments. Additionally, a heuristic, empirically motivated rule of ignoring the interest points from the top and bottom 1/6 of the frame is applied, as the top and bottom edge regions are more likely to contain non-sport-related content. These preprocessing steps aim at reducing the amount of interest points corrupted by device motion or background content.

2.3.2 Modality qualities

Modality data quality estimation is used as means for dynamic adaption in the fusion. The quality of the modalities is estimated from the sensor calibration level reported by the recording device, the noisiness of the audio, and the darkness of the video. As the magnetometer is sensitive to magnetic interferences, its calibration level is reported for each reading from the discrete set of values $\{0, 1, 2, 3\}$. This value is used as a quality estimate q_s for the sensor modality. For the audio quality a Gaussian mixture model (GMM) classifier is trained to distinguish audio frames containing people cheering close to the recording device, which was identified as the major source of noise corrupting

or in the worst case completely masking out the sounds from the events in the recorded sport. The audio quality q_a is then estimated as the ratio of non-cheering frames and the total amount of audio frames in the video clip. The video quality q_v is estimated as the brightness of the visual data. This is done in publication 2 by inspecting the V component of hue, saturation, value (HSV) color space and in publication 3 with a luminance measure L calculated from linearized RGB color components R,G,B as

$$L = 0.2126R + 0.7152G + 0.0722B. \tag{2.15}$$

Both methods result in a value between 0 and 255 with lower values indicating darker pixels. The values are calculated for each pixel of each analyzed video frame and averaged over all pixels and frames to get the visual quality estimate for the video clip. In publication 2 the qualities are quantized as low or high with the sensor quality being low with calibration value of 0 and high otherwise, the audio quality being low for ratios of 0.4 and above, and the visual quality being low for values of 70 or lower, i.e., $q_1^{\text{thr}} = q_a^{\text{thr}} = 0.4$, $q_2^{\text{thr}} = q_v^{\text{thr}} = 70$,$q_3^{\text{thr}} = q_s^{\text{thr}} = 1$. In publication 3 the quality values are normalized to the range from 0 to 1 and the thresholds optimized by grid search for each examined fusion strategy. Additionally, the average of q_v and q_s is used as the quality estimate q_t of the spatio-temporal video modality, since information from both the video data and the sensor signals affect the calculation of the sensor-enhanced STIP features.

2.3.3 Fusion and video-to-event aggregation

In publication 2 four late fusion approaches are compared: majority voting of the modality predictions, weighted average of likelihoods, and two versions of SVM fusion of likelihoods. All fusion methods incorporate simple dynamic modality adaption schemes from quality estimates of the data of all three modalities from the video clip being classified. Publication 3 considerably extends the set of considered fusion approaches from those of 2 by introducing the new STIP modality, and incorporating early, hierarchical, and additional late fusion schemes.

Early fusion

The early fusion is done by concatenating the features of all modalities and training a linear SVM on the concatenated data. The sensor features and STIP features are extracted from the whole video, whereas the audio and global spatial video features are extracted on each considered frame, so the frame-wise features are averaged over the whole video clip prior to the concatenation in order to deal with the asynchrony problem. In the early fusion case the modality qualities are used for scaling the features. Normalization of the ranges of different dimensions in a training data set is a standard procedure in order to avoid a certain dimension dominating the distance calculations in SVMs and many other machine learning tools that are based on calculating magnitude-sensitive distances [63]. This property is exploited by scaling down the features corresponding to a given modality, if the quality of the modality data is estimated bad. Similar ideas have been proposed on a single feature dimension basis generalizing feature selection – i.e., hard weighting with binary weights $w_k^b \in \{0, 1\}$ – to soft selection using continuous weights $0 \leq w_k^c \leq w_{\max}$ [64–67]. However, here the scaling is applied on per-modality basis, which avoids the relevance estimation for each single feature element as well as modifications to the SVM training criterion. Additionally, the scaling is only applied if the quality q_{ik} of a modality k in recording i drops below an empirically set threshold

value q_k^{thr}. The scaling coefficient σ_{ik} for the features F_{ik} ranges from 1 to 0 and depends linearly on the ratio of the quality value q_{ik} of and the threshold q_k^{thr} conditioned on crossing the threshold:

$$F_{ik}^{\sigma} = \sigma_{ik} F_{ik}, \tag{2.16}$$

where

$$\sigma_{ik} = \begin{cases} 1 & \text{if } q_{ik} \geq q_k^{\text{thr}}, \\ q_{ik}/q_k^{\text{thr}} & \text{otherwise.} \end{cases} \tag{2.17}$$

Hard thresholding was also experimented with, i.e., setting σ_{ik} as 0 for values below the threshold, but this gave unsatisfactory results in preliminary tests, so it was discarded at an early stage over scaling.

Hierarchical intermediate fusion

The classification framework originally proposed in [68] for combining different features or their subsets is adapted into a multimodal context as a hierarchical intermediate fusion approach. It trains one-against-all binary classifiers separately for all the modalities and feeds their outputs to one-against-all fusion classifiers. The most confident class-wise prediction is picked as the overall fused output. SVMs are used as the binary classifiers in the system. The main difference between this scheme and late fusion is that the fusion of different modalities is done explicitly separately for each class before fusion between the multimodal single-class predictions.

Late fusion

Prior to the late fusion approaches the following classifiers are trained for the different modalities: SVM with a polynomial kernel for the global spatial visual features, SVM with a linear kernel for the STIP features (in publication 3), SVM with a RBF kernel for the sensor modality, and GMM with 16 Gaussian component densities for each sport type for the audio modality. The SVM kernels have been chosen with cross-validation, and output probabilities are estimated from the predictions using the method presented in [54].

In majority voting fusion the qualities are used to exclude low-quality modalities from the voting. In weighted average of likelihoods as well as the first SVM fusion type of publication 2 the qualities are optionally used to modify the unimodal output likelihoods prior to the fusion stage. Specifically, the unimodal likelihoods $P_k^{\sigma}(C_j|v_i)$ for video v_i belonging to class $C_j, 1 \leq j \leq N_c$ are weighted towards uniform likelihoods $P_u = 1/N_c$ to de-emphasize the contribution of the modality:

$$P_k^{\sigma}(C_j|v_i) = \sigma_{ik} P_k(C_j|v_i) + (1 - \sigma_{ik}) P_u. \tag{2.18}$$

Thus, the weighted average fusion score S_{ji} and quality-modified weighted average fusion score S_{ji}^{σ} are calculated for class C_j and video i by equations (2.19) and (2.20), respectively, with the modality amount K being equal to 3 for publication 2 and 4 for publication 3.

$$S_{ji} = \sum_{k=1}^{K} w_k P_k(C_j|v_i) \tag{2.19}$$

$$S_{ji}^{\sigma} = \sum_{k=1}^{K} w_k P_k^{\sigma}(C_j|v_i) \tag{2.20}$$

The weights w_k of the modalities are derived as the cross-validated video-level classification accuracy of each modality. In majority voting the modality video-level accuracy ranking from the best to worst is used for handling situations with no majority – i.e., cases of disagreement between all the high-quality modalities included in the voting – by choosing the decision of the most accurate single modality that has high-quality data for the classified clip. The unimodal accuracy rankings are audio, video, and sensors in publication 2 and global spatial video, audio, STIP, and sensors in publication 3. The differences in the rankings are due to manually refining the evaluation dataset by removing some videos with non-sport related content between the publications. In the second SVM fusion type of publication 2 the non-thresholded qualities are normalized and concatenated with the unimodal likelihoods as additional input to the SVM. Additionally, in publication 3 the hierarchical fusion scheme of section 2.3.3 is applied in a late fusion context by using the unimodal likelihoods as input instead of the features.

Video-to-event aggregation

After obtaining all the video-level predictions for a certain event, they are aggregated as a single prediction for the event by majority voting. Additionally, in publication 3 majority voting is also applied directly to the pool of unimodal predictions of all video clips of the event, i.e., without multimodal fusion of the video-level predictions. This method is referred to as 2D majority voting. As with the two-stage majority fusion, this method is examined both with and without quality thresholding based modality excluding.

2.3.4 Experimental results

The sport type classification task was evaluated on a dataset recorded in sport events of six different types: soccer, american football, basketball, tennis, ice-hockey, and volleyball. A dedicated recording software was used for capturing the device motion sensor data along with the video. Altogether 507 videos with total duration of 73 hours and 2 minutes were recorded by 112 users in 22 sport events. The recording sessions were arranged so that multiple users were recording the same event from different viewpoints. For publication 3 the dataset was refined by manually removing videos with non-sport content, which resulted in a set of 479 videos spanning 68 hours and 44 minutes. The performance was assessed with classification accuracy using leave-one-out cross-validation. Specifically, full events (i.e., the set of videos shot at the event) were left out at each cross-validation iteration instead of single videos. As the work of publication 2 is extended in 3 (with minor differences in the used dataset) only the results of the latter are summarized here. Table 2.3 shows the unimodal accuracies on video ($\mathrm{Acc_V}$) and event level ($\mathrm{Acc_E}$), i.e., correctly classifying each separate video, and aggregating the predictions of videos from a certain event for classifying the event, respectively. The performance of the multimodal fusion with the different fusion approaches is shown for all combinations of two modalities in Figure 2.3 and for combinations of more than two modalities in Figure 2.4. The fusion

Table 2.3: Accuracies of the separate modalities on video (Acc_V) and event level (Acc_E). Reproduced with permission from publication 3.

Modality	Acc_E (%)	Acc_V (%)
Sensor	54.55	34.31
Audio	72.73	67.07
Spatial visual	81.82	68.45
Spatio-temporal visual	81.82	53.61

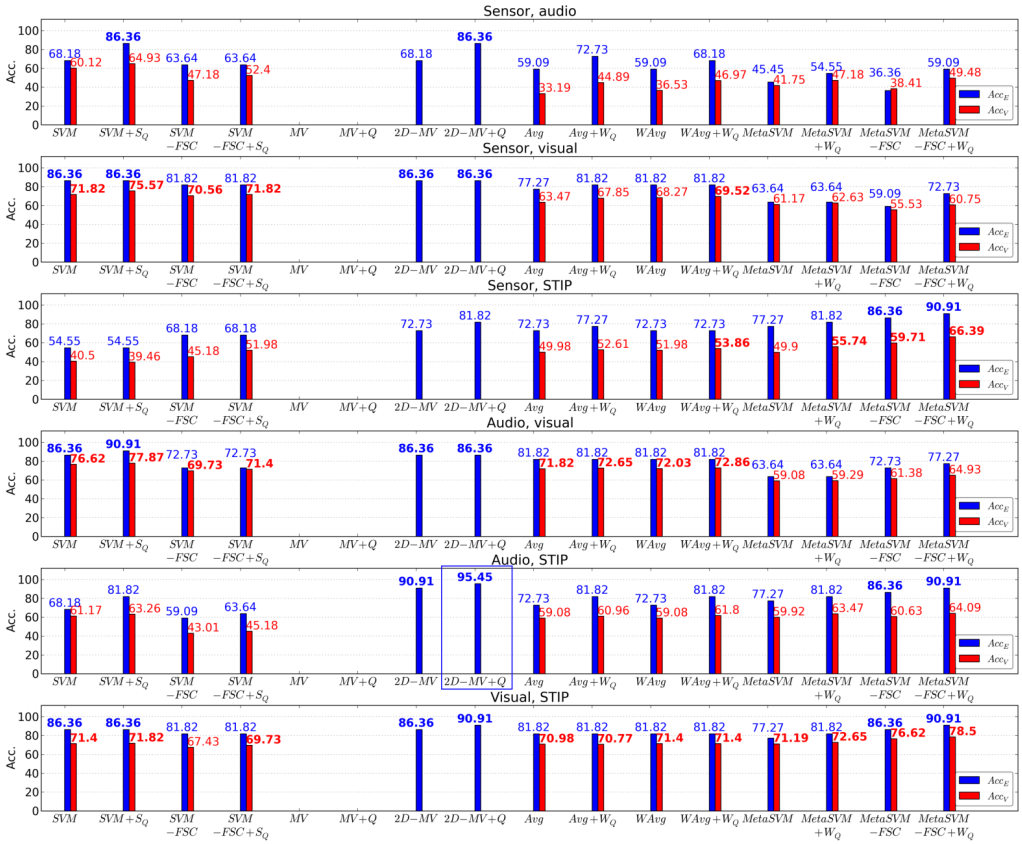

Figure 2.3: The subplots show the fusion results between all combinations of two modalities for all the considered fusion methods. The combinations improving the unimodal accuracies of the components are shown in bold. The blue rectangle highlights the best event-level accuracy. Reproduced with permission from publication 3.

method abbreviations used in Figures 2.3 and 2.4 are explained in Table 2.4. More details on the results can be found in publications 2 and 3.

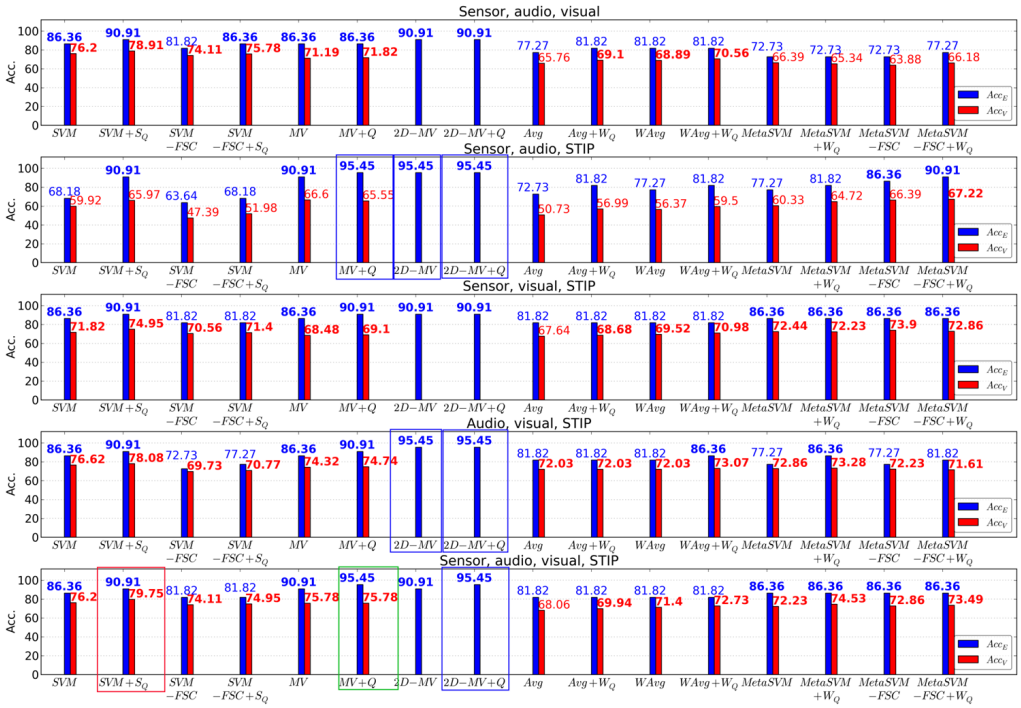

Figure 2.4: The subplots show the fusion results between all combinations of three and four modalities for all the considered fusion methods. The combinations improving the unimodal accuracies of the components are shown in bold. Blue and red rectangles highlight the best event- and video-level accuracies, respectively, whereas the green rectangle highlights the best pair of event- and video-level accuracies. Reproduced with permission from publication 3.

Table 2.4: Explanation of fusion method abbreviations used in Figures 2.3 and 2.4. Reproduced with permission from publication 3.

Fusion method	Abbreviation
Early fusion	
SVM	SVM
SVM using qualities for weighting features	$SVM + S_Q$
Intermediate fusion	
SVM fusing separately for each class	$SVM - FSC$
SVM fusing separately for each class and using qualities for weighting features	$SVM - FSC + S_Q$
Late fusion	
Majority voting	MV
Majority voting using qualities to select modalities	$MV + Q$
2D majority voting	$2D - MV$
2D majority voting using qualities to select modalities	$2D - MV + Q$
Average	Avg
Average using qualities for weighting probabilities	$Avg + W_Q$
Weighted average	$WAvg$
Weighted average using qualities for weighting probabilities	$WAvg + W_Q$
Meta-level SVM	$MetaSVM$
Meta-level SVM using qualities for weighting probabilities	$MetaSVM + W_Q$
Meta-level SVM fusing separately for each class	$MetaSVM - FSC$
Meta-level SVM fusing separately for each class and using qualities for weighting probabilities	$MetaSVM - FSC + W_Q$

3 Modeling cut timing of concert videos

A common goal for technical advancement in many creative fields is streamlining processes with a human in the loop to save the time and energy of the person to the most essential and interesting parts of the process. The human can even be excluded from the process altogether in case of uninteresting routine tasks. The streamlining can be achieved by means of automation and abstraction. As a side effect of this (along with ever decreasing equipment costs), the process often becomes more widespread among people with no time or interest to familiarize themselves with the previously laborous or repetitive parts of the task. This phenomenon is clearly observable for example in photography, videography, and music authoring, thanks to steadily improving quality of capturing and processing devices with intelligent, automatic tools for adjusting the device properties to best suit the capturing conditions or to achieve the desired feel for the end result. Examples of such abstraction tools include autofocus, automatic exposure bracketing, content-based matching and analysis of multiple images for automatic image compositions (e.g., panorama stitching and multi-exposure high-dynamic-range (HDR) imaging with standard sensors), optical and digital video stabilization, automatic gain control, automatic pitch correction (e.g., "autotune"), software emulation of musical instruments, etc.

Often the goal of the technological abstraction is to hide unnecessary technical details and routine decisions and leave the human in charge of the actual creative process with as intuitive tools as possible. Yet, the artistic task itself often requires vision and craftsmanship only achievable with dedicated practicing of the craft regardless of the abstraction provided by the tools. This still limits some artistic processes to indiduals, who are motivated enough to take the time to study the craft. As an example, a person with a modern digital camera or mobile device can effortlessly record video of high technical quality, yet aesthetically pleasant composition or editing the raw recordings to resemble professional television or movie productions is a much more demanding task. In recent years, there has been increasing research interest for investigating computational creativity with a dedicated conference organized since 2010[1]. This chapter presents a system proposed in publication 4 for automating a specific creative task in live concert video editing: choosing aesthetically appropriate video shot cut timing in relation to the background music.

Temporal structuring of visual events is a natural and subconscious property of the human perception, as confirmed, e.g., by functional magnetic resonance imaging (fMRI) measurements by Zacks *et al.* [69]. They conducted a study, where the participants were shown unedited videos of four different daily activities: making a bed, doing the dishes,

[1] http://www.computationalcreativity.net/

fertilizing a houseplant and ironing a shirt. The participants were asked to first watch the videos passively, then to search actively for coarse temporal segmentation to natural and meaningful units, and at the last viewing round to try to segment the videos to finer segments. The study finds notable neural activity during perceptual event boundaries both in passive and active watching. The finer segmentation is also aligned with the more strongly observed coarser one.

In music videos the temporal visual structure typically has correlations with the structure in the music. Gillet *et al.* [70] investigate such audiovisual correlations between shot boundaries, visual motion, note onsets, and music section changes in music videos. They observe that the correlation strength is heavily dependent on the content and composition type of the music videos, e.g., videos showing the musicians performing tend to have stronger audio-visual correlations than videos depicting narrative content. They also report that in correlation-based audio retrieval shot boundary correlations work better than motion-based correlations, and shot correlation with onsets is more usable than with music section boundaries.

Zettl [71] describes different composition types in professional video production. Concert video productions are expected to be constructed mostly to follow a *homophonic* structure, i.e., with the audio content semantically matching the video content. Furthermore, in professional concert videos the video and audio also typically have a *literal* relation, i.e., the dominant sound sources are highlighted in the video, e.g., by showing a close-up of a band member performing a solo. In contrast, non-live music videos in general enjoy more artistic freedom in choosing between *non-literal* homophonic material, e.g., showing scenery reflecting the music mood or lyrical themes or people dancing to the music, or using entirely *polyphonic* audiovisual structure, where seemingly unrelated video and audio are combined. The compositional and structural dependencies have both advantages and disadvantages for automatic concert video analysis for computational creativity tasks. On one hand the two modalities could be used to assist the analysis of each other, such as boosting beat tracking by analyzing the rhythm of stage lighting. On the other hand also choosing an appropriate video view from multiple parallel recordings is more critical to avoid confusion caused by showing visual content contradicting with the currently dominant sound sources.

Due to the typical literal and homophonic structure, live music video recordings should generally maintain the temporal synchronization between audio and video, and usually the music should be kept continuous with the exception of summarization-type of tasks. It might thus seem that editing of personal live music video recordings would often be infeasible or only limited to segments of non-literal relation (e.g., audience shots), unless some artificial or non-realtime material – such as photographs or older videos of the artist – is mixed with the live video. However, by combining multiple recordings of a common performance, the editing can be done between the recordings to emulate a professional multi-camera production without breaking the audiovisual synchrony. Such multi-device recordings can be relatively effortlessly obtained by gathering multiple people to record the performance, or by searching recordings from web video and social media services in case of larger public events.

3.1 Related work

Modeling of concert video content and editing can be done by considering various properties of the audio and video modalities, and different relations between these

properties. Automatic analysis and consequent synthesis of various aspects of music videos (including live concert videos) has been studied in the literature. Such analyses include for instance the detection of lyric sentence boundaries [72], music structure [72], applauses [73], significant moments [73, 74], and instrument solos along with classifying the solo instrument [73], as well as distinguishing between vocal and instrumental parts [75]. Different audiovisual relation observations in the related literature include visual content of music videos being repeated based on musical structure [76], i.e., using similar shots or shot sequences every time a certain musical part is repeated, matching video cutting frequency with music tempo [77], aligning video shot switches with strong background music beats [77], and matching video motion intensity with the tempo of the music [77]. In [78] it is proposed to match the degree of audio energy change with video movement, video brightness with the spectral centroid of audio, and the lengths of different audio and video segments. The authors form a cost function for the matching degree of audio and video segments as a linear combination of the matching rules. In [79] sudden visual changes (e.g., sharp cuts) are matched with sudden audio changes and similarly gradual changes in video with smooth changes in audio. It is argued in [75] that the audio analysis should adapt to different music genres, and according to Liao *et al.* [80] analysis and matching might be sensible to be done separately for related subsets such as performances from one artist, venue, or director for more interesting audiovisual patterns.

Snoek *et al.* [81] examine the applicability of visual-only content and style analysis on concert video indexing. They use 12 visual detectors for typical semantic concepts in concerts – namely *audience*, *band*, *drummer*, *face*, *guitarist*, *instrument*, *keyboard*, *person*, *rear-view*, *singer*, *stage*, and *turntable*. The concept detectors are combined with shot-size (e.g., wide, medium, close-up) analysis from the sizes of detected faces, when available, detection of camera operations, as well as overall estimation of camera motion. However, it is not trivial to robustly distinguish from mere video content whether a shot is recorded near the subject with a wider-angle lens or from further away with a longer lens. Classification of camera operations as being static, panning, tilting, zooming in/out etc., has commonly been carried out based on the video content (see, e.g., [82, 83]). In some use cases an alternative, often more efficient approach is to estimate the occurrences of the operations (with the exception of zooming) from the auxiliary sensors embedded in most modern multimedia devices as discussed in chapter 2. In [84] the auxiliary sensor signals from multiple mobile recording devices are used collectively to estimate the area of interest in an event from the intersections of the device pointing direction vectors with the assumption that most people recording an event tend to point their recording devices towards the interesting targets. Wang and Cheong [85] define framing and motion based cinematographic directing rules in the context of movies for the purpose of shot indexing and retrieval.

Matching the semantic content of the audio and video can aid in creating a more pleasant automatic edit e.g., by showing a close-up of an artist singing or the crowd when cheering is heard [86]. During instrument solos it would make sense to show imagery of the solo artist, even though Naci [73] argues that this is not always done by professional directors. Solo sections can naturally be accompanied by visuals of the audience, wide angle shots of the stage, but for instance zooming in on a bass player during a guitar solo can easily be interpreted as a mistake from the director or editor. Further, different analysis approaches within one modality (such as visual content vs. visual style) can vary in performance for different analysis tasks as argued in [81]. Crowdsourcing different phases of concert video production has also been experimented with, e.g., by virtual director assisted collaborative concert recording [87] as well as for cutting and view choosing [88].

Automatic song segmentation and identification from live concert recordings has lately been studied both with audio-only [89] and audiovisual [90] approaches. Automatic detection of shot boundaries in edited videos is an established branch of video analysis research [91]. Live concert videos pose a challenging scenario for shot boundary detection, as camera flashes from the audience, stage lighting, and other prominent augmented visual elements, such as video screens, can easily cause false shot boundary detections. Technical quality of user-generated videos has been addressed by using *no-reference* or *blind* quality measures [86] – such as blockiness, blurriness, brightness, and shakiness – as well as filtering out segments with excessive camera motion [76, 92], low color entropy [76], or poor contrast [92].

Postprocessing effects are often used to enhance the look and feel of professional concert video productions. Visual time-scale modification effects such as speeding up and slowing down some parts of the video have been applied for enhancing user-created video interest [78, 93]. The use of these effects is more limited in the music video domain due to audiovisual synchronization requirements from the homophonic structure and literal relation. However, when there is no direct literal relation observable (e.g., when not showing the artists or showing them from afar), time-scale modification can be used for improving the aesthetic connection of audio and video for instance by applying it in synchronization with certain audio events.

Many of the music-enhanced editing approaches found in the literature are tuned for either matching separate background music to video or forming a video sequence, to which audio is matched afterwards. Both of these approaches are impractical for editing concert videos. In contrast to generic automatic music video generation works, in the concert video editing setting, the temporal matching of audio and video is fixed and the content of different candidate clips from different recording devices generally has less variation.

3.2 Cross-modal dependencies between music and video

As discussed in chapter 1, multimodal analysis can be divided to multimodal fusion, which deals with improving performance on a task by jointly utilizing multiple modalities, and cross-modal processing, where dependencies and multimodal patterns are sought between the modalities. By this categorization, modeling the timing of shot cuts in concert videos with regard to the music content is clearly a cross-modal processing task (although multimodal fusion might also be well applicable to some subtasks). According to Shivappa *et al.* [6], in human development cross-modal correspondence learning starts at an early age and is used to combine multimodal information at various levels of abstraction in combination with other techniques. Below some common approaches to cross-modal processing are discussed with example works from the literature presented in the context of music video analysis and synthesis.

One approach for modeling cross-modal dependencies is by learning a mapping to a latent space, where the different modalities can be directly compared. Liao *et al.* [80] propose to mine audiovisual patterns in music videos using a dual-wing harmonium model, which is a bi-modal extension of the RBM with the nodes of both modalities being connected to a common set of latent nodes. CCA is extended in [94] to ranking canonical correlation analysis (R-CCA) and multiple ranking canonical correlation analysis (MR-CCA) for handling pairwise ranked instances of modality combinations in multiple clusters. The methods are applied to the task of mapping the content and descriptive tags of music to the visual content of music video key frames. The clustering is based on the audio

modality and the modality representations are transformed as distances to randomly picked templates in the clusters prior to applying CCA and its extensions.

Heuristic rules can also be used for matching audio and video. Mulhelm *et al.* [95] propose matching the two modalities by transforming them with hand-engineered linear projections to a common pivot space. The pivot space transforms are based on a heuristic mapping between video and audio aesthetic features proposed in [71]. Example aesthetic feature pairs from the mapping include color hue and sound pitch, color saturation and sound timbre, as well as motion vectors and music tempo. The aesthetic feature values are transformed to fuzzy membership values of non-overlapping low, medium, and high value ranges prior to the pivot space projections. Yoon *et al.* [78] describe a system for automatic music video generation by matching perceptually similar video and audio segments. They segment the input audio and video according to audio novelty analysis and video similarity changes. Video segments are then matched to each audio segment based on rules such as matching the video and audio segment boundaries, matching the degree of audio energy change with video movement, matching video brightness with the spectral centroid of audio, and matching the lengths of different segments. Based on a cost function, which is a linear combination of the rules, a video segment is chosen for each audio segment and the video is time-stretched to match the duration of the corresponding audio segment.

Besides matching the modalities by heuristics or in a latent space, the use of various supplementary, intelligible properties has been examined in the literature for establishing audiovisual dependencies. Wang *et al.* [96] map audio and video of music videos to a three-dimensional emotion space according to estimated *valence* (i.e., degree of pleasantness or unpleasantness), *activation* (i.e., the intensity), and *potency* (i.e., degree of dominance or submissiveness) factors. In [97] the audio content is analyzed to estimate the music mood in a simple scale from positive to negative. The mood information is then used along with lyrics for retrieving still images for music slideshow video generation. Lyrics are also used in [98] for retrieving images with text tags. In [99] the lyrics-based retrieval is extended by content-based matching of the tagged public images with untagged images from personal collections for personalized music slideshow video generation. Although such supplementary properties may offer means for revealing dependencies that would otherwise be difficult to observe, the properties might not always be available (e.g., lack of lyrics in instrumental music or their unknown time-alignment in live recordings due to unknown song starting points with regard to the beginning of the video recording as well as tempo variations and other changes in live performances) and the nature of the dependencies might be biased to domains, where the given property is inherently used. Additionally, the use of such properties introduces the burden of estimating, retrieving, measuring, or annotating the properties. Naturally, different cross-modal matching approaches can also be combined as hybrid systems. Wu *et al.* [100] propose to use lyrics-based image retrieval to complement retrieval using MR-CCA mapping between audio and images.

Regardless of the used approach, mapping visual content with music is challenging due to the highly subjective nature of the problem, as an audiovisual connection obvious for one person can be highly contradictory for another. Yet, dependencies of some directly comparable properties such as matching the tempo of music to the rate of a temporally repeating visual element are quite universally acknowledged. Automatic visual rhythm estimation in video is investigated in [101]. Visual rhythm can arise e.g., from person movements, camera operations, or lighting changes. The authors use absolute frame

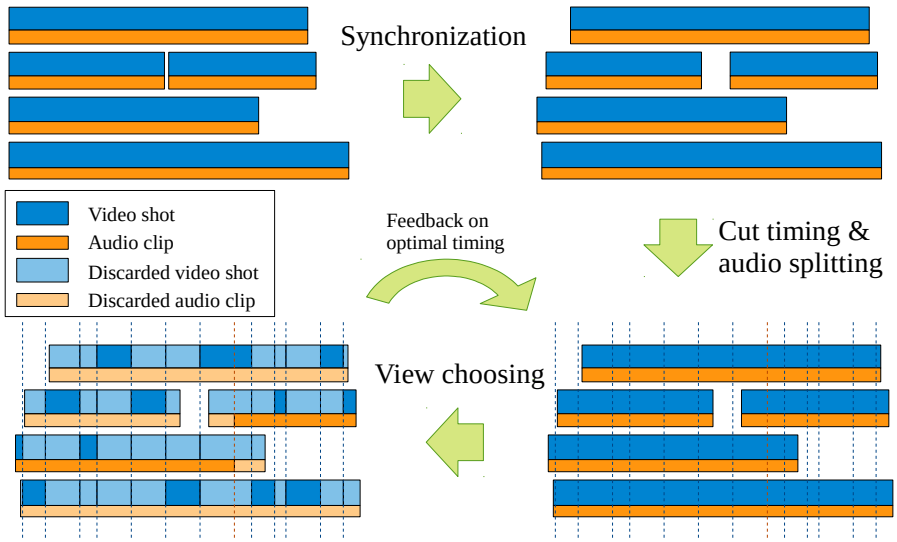

Figure 3.1: In multi-camera mashup creation the videos and the corresponding audio tracks from the different recording devices need to be temporally synchronized, suitable cut times assigned in the common synchronized time frame, and the optimal views and audio clips chosen from the set of available views and audio tracks for each shot. The choice of view may affect the optimal cut time.

difference and 2D angle-magnitude histogram of *optical flows* to form a novelty score, from which visual tempo is estimated by autocorrelation. Two example applications are presented: modifying music tempo to follow the estimated visual tempo, and creating custom music videos by time-stretching video with visual rhythmic elements to match the tempo of a song. The correlation of music and visual rhythm in dance videos is studied in [102]. The visual rhythm is estimated by detecting changes of direction and pauses in the trajectories of tracked visual keypoints. Besides the single-camera approach, audiovisual correlation analysis of dancing has also been conducted using dedicated multi-camera setups [103] and commercial off-the-shelf RGB-D sensors [104].

In this chapter, rather than trying to find dependencies between music and low- or high-level content depicted in visual frames, patterns are sought between music meter and the timing of shot cuts based on the analysis of professionally edited concert videos. The goal is to predict appropriate cutting times for unedited user-recorded concert videos from the estimated music meter, which avoids the need for visual content analysis in the synthesis phase. The work has been developed for the application of automatic multi-camera mashup creation, which is briefly covered in the following section.

3.3 Multi-camera mashups

Timing of cuts between different videos is a central problem in automating the creation of so called multi-camera mashups. Multi-camera mashup is a term used for a single

video formed from a set of user-generated video clips recorded concurrently with multiple devices in a common location, by switching between the different views offered by the different recording devices. The aim is to make the resulting mashup video resemble professinally recorded and edited videos by optimizing the view choosing and timing of shot switches in terms of content quality, relevance, view variety, and cinematographic principles. In case of music event recordings, also cues from the music can be utilized for the editing. Multi-camera mashups are distinguished from video mashups [105, 106], remixes [79], or montages [107] by the use of videos with a common timeline recorded with multiple recording devices in a common location instead of arbitrary, possibly unrelated source video and audio material combined without timeline limitations. Figure 3.1 shows the main phases of a multi-camera mashup generation task. Although the shot switches and audio clip changes are depicted as sharp cuts, they can also be longer duration transitions (e.g., by cross-fading or other effects). As the audio of edited videos – especially in music and concert videos – rarely shows spatially jumpy cutting behaviour, the switching between the different audio tracks is usually kept to a minimum. A brief overview of multi-camera mashup systems with different degrees of task automation is presented below, highlighting the cut timing aspects of the presented approaches as the work described in this chapter concentrates on the cut timing problem.

Shrestha *et al.* [86] present a system for automatic multi-camera mashup creation from videos recorded in concerts. The automatic editing task is formulated as an optimization problem over a linear combination of a set of user requirements for a pleasing mashup video. This avoids the use of predefined fixed editing rules and provides means for emphasizing the different requirements by adjusting their respective weights. They synchronize different videos using audio fingerprints, assess image quality and diversity from video content, estimate cut point suitability from camera motion and brightness change analysis of video as well as manual annotations of audio percetual changes (after unsatisfactory results from a beat and tempo detector), and finally greedily maximize their objective function, which formalizes the user requirements and constraints. The timing of shot cuts is based on the cut point suitability scores and fixed genre-dependent maximum and minimum shot length thresholds.

Saini *et al.* [108] describe a system for creating mashup videos from user-contributed unedited content from live performances. They propose to choose each shot in the mashup based on the combination of view quality filtering and keeping track of the history of previously selected shots for view diversity. They analyze professionally edited videos to learn shot transition and shot length distributions between classes of different view distances and directions in relation to the captured performance. After view quality filtering the input videos are classified as center, left, and right views as well as near and far views. These classes are used as the states of a finite state machine. A HMM is formed from the state transition probabilities and a shot length emission matrix for generating shot state and length sequences. At each time the videos of the current state are ranked according to their visual quality and view diversity compared to the previous shot. The highest ranked video is selected as the next view and its duration defined based on the learned length distributions with the length altered according to video quality. Although their system is aimed at live performances, it is based only on the visual information omitting any analysis of the audio modality.

In [109] Arev *et al.* describe a mashup creation system for synchronized footage from wearable or hand-held cameras capturing a joint activity of a group of people. The authors use the centers of attention of the individual cameras to estimate the important

regions in the scene. This is done by content-based camera motion trajectory estimation with a structure-from-motion technique. Besides view quality, variety, and the important region estimation, the 180 degree rule and avoiding of jump cuts are used as guidelines for the view choosing. The 180 degree rule states that motion in the scene should not reverse direction due to view switching, i.e., the view should be switched to a camera at the same side as the current one with respect to the direction of the motion. Jump cuts occur, when a view switch is made to a camera that is too close to the current view. This can easily be perceived as a glitchy artefact instead of a proper view switch. In addition to enough variance in the shooting direction of concurrent shots, also diversity in the shot framing size (e.g., close-up, wide-angle) is encouraged. The system optionally crops the shots centered to the estimated important region for additional shot size variations. The timing of cuts and view choosing is based on optimizing a path through a graph, where the nodes represent the different cameras at a given time instant and the edges the costs of switching to the camera at the other end of the edge. Each node also has a cost calculated as a weighted average of various view quality properties. Shot durations are limited by minimum and maximum lengths. They also experiment with cutting on notable action, i.e., when the joint center of attention of the cameras shifts abruptly.

Bano and Cavallaro [110] describe a framework for multi-camera mashup creation from synchronized user-recorded videos of a common event. Spectral rolloff is extracted from non-overlapping 1 second frames from the overlapping temporal segment of the audio tracks of the separate videos and the audio tracks ranked according to their frame-averaged spectral rolloff with the assumption that low spectral rollof corresponds to higher quality audio with less high-frequency noise. A single audio track is then stitched from the alternative audio segments available at any given time based on the quality analysis. The stitched audio track is further analyzed for three low-level features – root mean square (RMS) intensity, spectral centroid, and spectral entropy – and cut points assigned according to co-occuring changes in the features along with maximum and minimum shot length limits. The changes in the chosen low-level features are assumed to result from various higher-level changes such as instrumentation variations in music or the audio content switching between speech and music. Visual spatial frame quality and spatio-temporal camera motion stability analysis as well as view diversity against the two preceding shots are used for view choosing. Visual content similarity clustering is also experimented with for avoiding jump cuts.

The multi-camera mashup system by Wu *et al.* [111] provides means for video synchronization by audio fingerprints, audio and video quality assessment, view diversity and filming principle constraining, and separate optimization for cutting points of the video and audio. The editing rules and guidelines for the system are based on a focus study with video editing professionals. The cut timing is based on minumum and maximum shot duration, adjusting the cutting rate according to audio tempo approximated from the audio onset rate, and encouraging cuts during speech or singing pauses estimated from the audio energy. The video shots are segmented to subshots by color and motion change detection. Camera choosing is then done according to the combination of subshot-level spatial and spatiotemporal quality inspection, content diversity of adjacent shots to avoid jump cuts, and the requirement of static cameras during cuts. Additionally, they try to balance between discarding low-quality audio segments and minimizing the amount of audio source switches.

The work described in this chapter offers an alternative approach to the cut timing subproblem in the context of music events. The approach delves deeper into music-specific

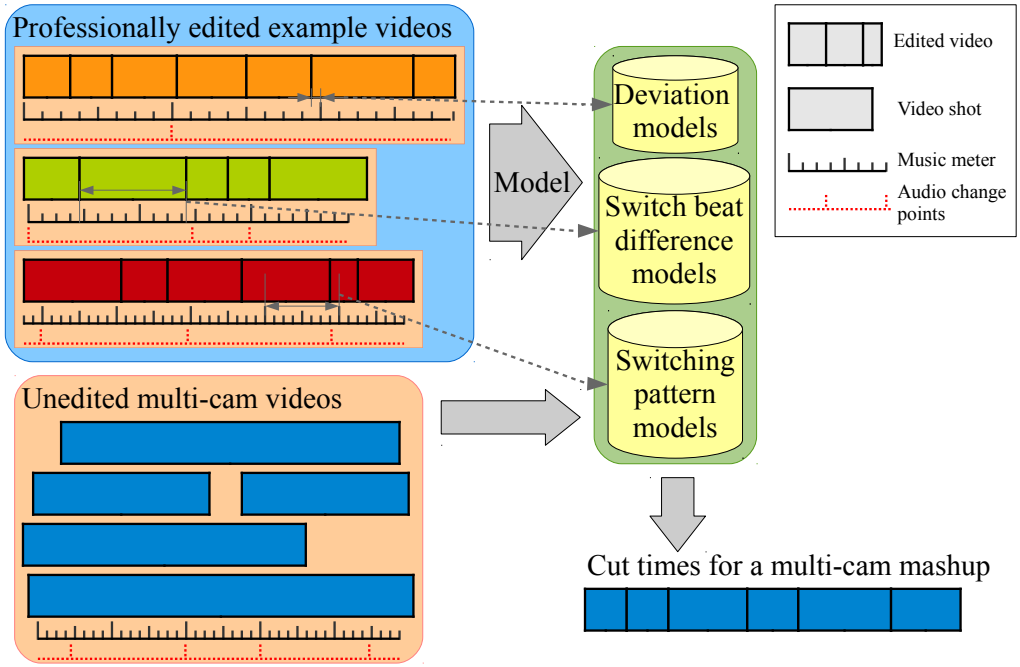

Figure 3.2: The cut timing of professionally edited example concert videos is analyzed with regard to music meter and audio change points. The deviation models are formed from the relative time differences between the cuts and the closest beats. The switch beat difference models capture the information about the typical shot lengths in beats. The switching pattern models are formed from the cut timing patterns occurring within two-bar segments. The models are used to synthesize cut times for a set of multi-camera concert videos based on the music meter and audio change points analyses of their common synchronized audio track. Reproduced with permission from publication 4.

audio analysis in hopes for improved aesthetic connection between the cut timing and the music. Rather than using hand-defined rules, the system learns a model of audiovisual cutting patterns from example data.

3.4 Cut timing modeling and synthesis

The art of music video cutting is constant interplay between different factors such as the music mood, tempo, as well as the acoustic and visual events. An intuitive and often used approach to automatic video cut timing is to model the length of shots from example data, which is suitable for non-music recordings as well as music events with loose connection between the music and editing or rhythmically complex cases. An alternative way for better matching the video editing with music is to analyze, how the switches are distributed over the duration of musical measures. The switches can be related to musical beats by quantizing the switching times to the nearest beat time.

This chapter presents an example-based cut timing modeling and synthesis framework aimed for automatic multi-camera mashup creation from concert events. The work was originally proposed in publication 4. Figure 3.2 shows a high-level overview of the modeling and synthesis processes. In the modeling phase a set of professionally edited concert

videos – hand-annotated with their shot switching times (i.e., the occurrence times of sharp cuts or the mid-points of gradual transitions) and music meter on beat-, bar-, and two-bar pattern level – are analyzed for switching patterns, switch beat differences, as well as deviations of switches from the beat times. Switching pattern modeling captures typical cut patterns in two-bar sequences as well as the tendencies of certain patterns to be followed by others. Switch beat difference modeling is conceptually analogous to shot length modeling done in the literature, but here the length is calculated as the amount of beats passed since the previous view switch. The switch beat difference model is used, when overriding cut times from the switching pattern model with cuts assigned to audio change points analyzed from the video under editing. By using the music meter as the temporal grid instead of absolute time, the modeling implicitly takes into account the tempo of the music. Switch deviation modeling analyzes how the exact cut times deviate from the closest beat times. The deviations are denoted as relative distances to the closest beats, i.e., ranging from -0.5 to 0.5 beats. The resulting models can then be used to generate cut sequences based on the common audio track of a set of multi-camera videos from a concert or other music performance.

3.4.1 Audio analysis

Different audio content analysis techniques are used in various parts of the framework. These include audio change point analysis, music section detection, and music meter analysis.

Audio change point analysis segments the input audio at points, where a notable change occurs in the content. The changes are analyzed using both MFCC and chroma features, with the former registering more generic changes in the sound spectrum and the latter changes in chord progression. Top-down iterative clustering is applied separately to both types of features in order to categorize the data to M different clusters. The cluster mean and variance vectors are used as the states of a fully connected HMM. After few iterations of training the model with the Baum-Welch algorithm and decoding the state sequence with the Viterbi algorithm, the audio change points for each of the two feature types are obtained as the points of state changes in the state sequence.

Music section detection uses the MFCC-based change points for segmenting the input audio to sub-parts of uniform content. GMMs trained for the classes music, speech, babble speech, and crowd noise are then evaluated on the segments. The segment is assigned to the class that gets the highest likelihood of the corresponding GMM having generated the features of the input segment. All sequences of consecutive segments with the highest likelihood for the music class are considered as music sections.

The music meter is analyzed on the beat, bar, and two-bar level. The time signature of the music is assumed to be 4/4, i.e., each bar consists of four beats, which is a reasonable assumption for analyzing the majority of popular music. With the assumed time signature the two-bar analysis divides the music to segments of eight beats used as the basic unit for cut timing modeling. The tempo of the input music is first estimated with the chroma-based method presented by Eronen and Klapuri in [112]. The beat tracking is then carried out with the dynamic programming routine from [113]. The estimated beat positions as well as the chroma features and chroma accent signal of the tempo estimation are used as input for locating the first beats of bars, i.e., the downbeats. The accent signal is sampled at the estimated beat positions and the samples concatenated into feature vectors of length four (for each beat in a bar in the assumed time signature). These features are then used for training a linear discriminant analysis (LDA) classifier

to distinguish between downbeats and other beats. The classifier produces a downbeat likelihood score sequence s_{db} for the estimated beats of a given song. Additionally, taking advantage of the fact that chords are often changed on downbeats, the chroma features are sampled on the estimated beat positions and the resulting sequence differentiated to get a chord change likelihood score sequence s_{cc}. The two score sequences s_{db} and s_{cc} are normalized over time and summed to obtain the downbeat likelihood signal S_{db}. The most likely downbeat sequence is then found by sampling S_{db} every four beats starting with different candidate offsets $\hat{o}_4$ from the first beat, and seeing which offset maximizes the average of the sampled signal. The beats at the sampling indices corresponding to this offset o_4 are predicted as the downbeats. Formally this can be expressed as:

$$o_4 = \arg\max_{\hat{o}_4} \frac{1}{N_{\hat{o}_4}} \sum_{n_4=0}^{N_{\hat{o}_4}-1} S_{db}(4n_4 + \hat{o}_4), \quad 0 \leq \hat{o}_4 \leq 3, \tag{3.1}$$

where $N_{\hat{o}_4}$ is the length of each candidate downbeat sequence.

The downbeats of the two-bar sequences are estimated in a fairly similar fashion with the addition of audio change point estimation. The LDA is trained to predict between two-bar downbeats and other beats, producing a two-bar downbeat likelihood score sequence s_{2b} from the chroma accent features of an input song. For the audio change estimation MFCC-based audio change points s_{cp} as well as audio novelty score s_{no} of [92] from beat-synchronous MFCC and chroma self-distance matrices are used. The motivation for including the audio change analyses is that the desired grouping of bars into groups of two should try to align the downbeats of the two-bar segments with music structure boundaries, which introduce changes in the music. Two-bar segment downbeat likelihood signal S_{2b} is formed by summing the temporally normalized score sequences s_{2b}, s_{cc}, s_{no}, and s_{cp}. Then two-bar segments are formed starting from the first and second bar, S_{2b} sampled at the indices corresponding to the first beats of the segments, and the more likely grouping chosen by maximizing the mean of the sampled signal. Formally, the offset o_8 in beats from the first beat is sought with the following equation:

$$o_8 = \arg\max_{\hat{o}_8} \frac{1}{N_{\hat{o}_8}} \sum_{n_8=0}^{N_{\hat{o}_8}-1} S_{2b}(8n_8 + \hat{o}_8), \quad \hat{o}_8 \in \{o_4, o_4 + 4\}, \tag{3.2}$$

where $N_{\hat{o}_8}$ is the length of each candididate sequence of downbeats starting a two-bar group.

3.4.2 Cut timing framework

The cut timing framework consists of an offline modeling phase and a synthesis phase for creating cut times for undedited concert video material. In the modeling phase professionally edited concert videos hand-annotated with music meter and cut times are analyzed for switching patterns, beat differences of cuts, as well as cut deviations from exact beat times. In the synthesis phase music sections are detected from the common audio track of a set of multi-camera concert videos, and the music sections analyzed for music meter and audio change points. Based on the analysis, the models created in the offline phase are consulted for suitable cut times. Figure 3.3 shows an overview of the cut timing framework.

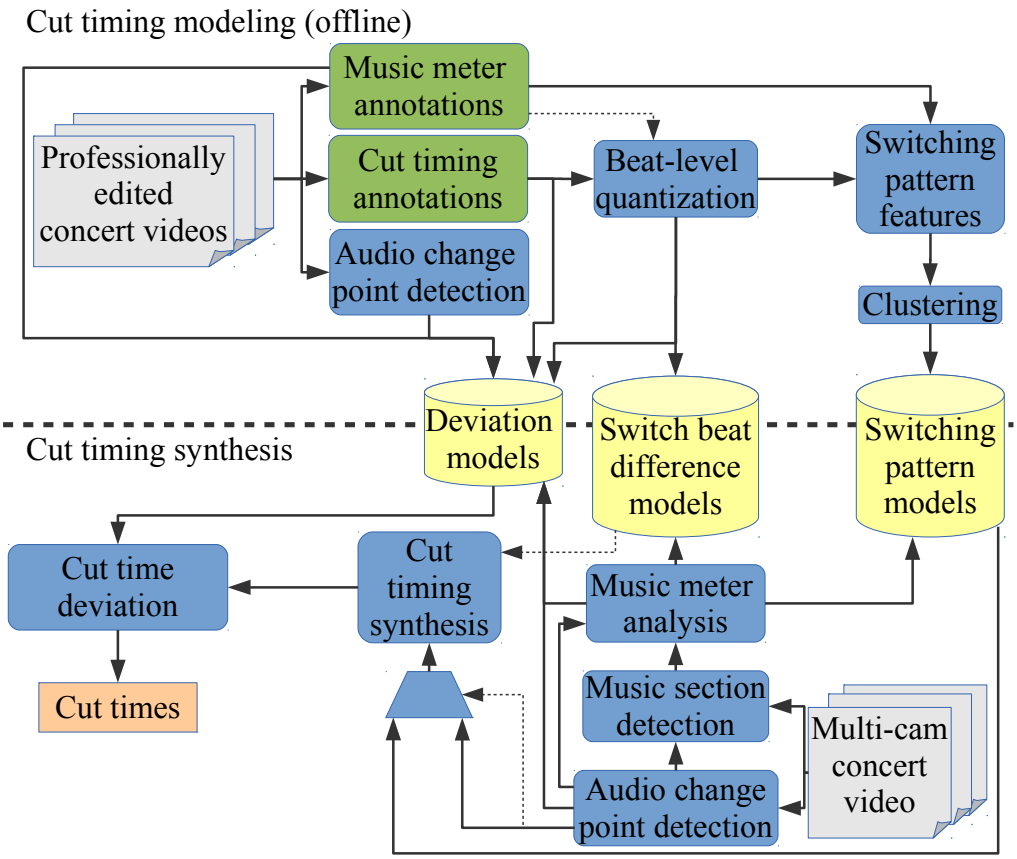

Figure 3.3: Block diagram of the cut timing modeling framework. Solid and dashed arrows indicate data flow and control signal, respectively. Control signal does not flow out of the controlled block. Reproduced with permission from publication 4.

The switching pattern models are formed from the two-bar segments of the professionally edited example concert videos. The cuts occurring within a segment are quantized to the closest beats forming a binary vector indicating the occurrence of cuts for each of the eight beats within the segment. As an example, a two-bar segment containing cuts closest to the downbeats of both bars results in a binary vector $[1, 0, 0, 0, 1, 0, 0, 0]$. To better capture the sequential relations of the cuts within a segment the binary vectors are further transformed into a beat difference representation by counting the beat difference between the cuts and padding to constant length with zeros as exemplified in Figure 3.4. The J videos in the example video set D are represented as sequences of the beat difference vectors $\mathbf{D}_j = [\mathbf{d}_{j,1}, \mathbf{d}_{j,2}, ..., \mathbf{d}_{j,N_j}], 1 \leq j \leq J$ with N_j indicating the amount of two-bar segments in the jth example video. k-medians clustering is applied on the example video set resulting in N_C cluster centers C. A quantized example set Q is formed by replacing each beat difference vector in D with the corresponding cluster center $\mathbf{c}_m \in C$ resulting in quantized sequences $\mathbf{Q}_j = [\mathbf{q}_{j,1}, \mathbf{q}_{j,2}, ..., \mathbf{q}_{j,N_j}], 1 \leq j \leq J$.

A switching pattern model is formed as a Markov chain (MC) model by setting each $\mathbf{c}_m$ as a state and estimating the state prior probabilities $P(\mathbf{c}_m)$ as

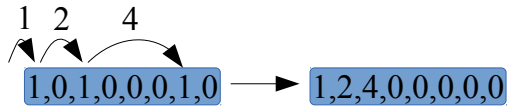

Figure 3.4: The beat difference feature representation is formed by concatenating the beat differences of the binary cuts-on-beat feature representation. Reproduced with permission from publication 4.

$$P(\mathbf{c}_m) = \frac{\sum_{j=1}^{J}\sum_{h=1}^{N_j}\mathbf{1}_{\mathbf{c}_m}(\mathbf{q}_{j,h})}{\sum_{j=1}^{J}N_j}, \quad \mathbf{c}_m \in C, \tag{3.3}$$

where $\mathbf{1}_{\mathbf{c}_m}(\cdot)$ is the indicator function of the argument being equal to $\mathbf{c}_m$. Transition probability $P(\mathbf{c}_n|\mathbf{c}_m)$ from state m to state n is estimated as

$$P(\mathbf{c}_n|\mathbf{c}_m) = \frac{\sum_{j=1}^{J}\sum_{h=1}^{N_j-1}\mathbf{1}_{\mathbf{c}_m}(\mathbf{q}_{j,h})\mathbf{1}_{\mathbf{c}_n}(\mathbf{q}_{j,h+1})}{\sum_{j=1}^{J}\sum_{h=1}^{N_j-1}\mathbf{1}_{\mathbf{c}_m}(\mathbf{q}_{j,h})}, \quad \mathbf{c}_n \in C, \mathbf{c}_m \in C. \tag{3.4}$$

The switching pattern model generates pattern sequences by drawing the initial pattern from $P(\mathbf{c}_m)$ and the consecutive patterns with $P(\mathbf{c}_n|\mathbf{c}_m)$.

Switch beat difference models capture the distribution of shot lengths in beats. In a video with T_j annotated beats and R_j cuts, for consecutive cuts r_j and $r_j+1, 1 \leq r_j < R_j$ with the closest beats t_{r_j} and $t_{r_j+1}, 1 \leq t_{r_j} \leq t_{r_j+1} \leq T_j$, the beat difference $\Delta(r_j+1, r_j)$ is calculated as

$$\Delta(r_j+1, r_j) = t_{r_j+1} - t_{r_j}, \quad 1 \leq r_j < R_j. \tag{3.5}$$

The switch beat differences of the example videos are aggregated to a cumulative histogram truncated at 24 beats and normalized to unity at the last bin. The histogram can thus be used to approximate the likelihood for a new cut after the amount of beats from the previous cut as indicated by the bin index. Separate cumulative histograms are formed from cuts closest to beats at each beat position within a two-bar segment, e.g., in case of the first beat position, including only cuts with t_{r_j} corresponding to a two-bar segment downbeat.

The motivation for the cut deviation modeling is to introduce natural variation to the beat-aligned cut times produced by the switching pattern models as well as the beat difference models. The deviation models are formed by histogramming the relative deviations of the cuts (ranging between -0.5 and 0.5 beats) from the closest beats prior to the quantization. Formally, if the times of cut r_j and the closest beat t_{r_j} are given by $c(r_j)$ and $b(t_{r_j})$, respectively, the relative deviation $\Gamma(r_j)$ is calculated as

$$\Gamma(r_j) = \begin{cases} \frac{c(r_j)-b(t_{r_j})}{2(b(t_{r_j}+1)-b(t_{r_j}))} & \text{if } t_{r_j} = 1 \\ \frac{c(r_j)-b(t_{r_j})}{2(b(t_{r_j})-b(t_{r_j}-1))} & \text{if } t_{r_j} = T_j \\ \frac{c(r_j)-b(t_{r_j})}{b(t_{r_j}+1)-b(t_{r_j}-1)} & \text{otherwise.} \end{cases} \tag{3.6}$$

Separate histograms are formed for each beat position. Additionally, all cuts are divided to those occurring closest to a beat, which is also the closest one to an audio change point, and other cuts, and separate histograms are formed for the two cases. After normalizing the histograms to sum to unity, they can be used for drawing cut deviations from the discrete set of the bin centers.

The models created in the offline modeling phase can be used for synthesizing cuts for new multi-camera recordings by detecting music sections and audio change points from the audio, the music meter from the music sections, and going through the music sections in two-bar segments $s_v, 1 \leq v \leq V$, where V is the amount of two-bar segments in the section. The processing of each two-bar segment is shown in Figure 3.5. For any segment with no detected change points, a switching pattern $\mathbf{q}_v \in C$ is queried from the switching pattern model given the previous switching pattern state of the model $\mathbf{q}_{v-1}$ according to the transition probabilities $P(\mathbf{q}_v|\mathbf{q}_{v-1})$ (or drawn based on the prior probabilities $P(\mathbf{q}_1)$ in case of the first two-bar segment). If the inspected two-bar segment contains a set of audio change points A_v, each $a_{vu} \in A_v, 1 \leq u \leq |A_v|$ is processed in temporal order as follows. Given the beat difference $\Delta(a_{vu}, r)$ from the previous cut time r to the change point a_{vu}, the switch beat difference model corresponding to the beat position of r gives a likelihood for assigning a cut on a_{vu}. If the beat difference from the preceding cut is too small according to the model, the cases of r being from the switching pattern model or due to an audio change point are handled differently. If r is also due to an audio change point, no cut is assigned. However, if r is from the switching pattern model, a cut is assigned on a_{vu}, and the cut at r is discarded. The likelihood-based assignment of cuts on audio change points retains the shot length distribution from the example data. Favoring the change point cuts over the ones from the switching pattern model emphasizes the audio content of the material to be cut over the statistical patterns learnt from the example data set. Whenever cuts are assigned on audio change points, the resulting two-bar cut sequence is used to update the previous state of the switching pattern model by finding the most similar pattern $\mathbf{a}_v \in C$ in the model. The similarity is checked iteratively by starting from the last (i.e., the latest) beat of the patterns and including more beats towards the beginning of the pattern. The state updating smooths out the transition from audio change point cuts to switching pattern model cuts. Finally, the deviation models provide discrete distributions for deviating the cuts from exact beat times according to the beat position and the cut type (switching pattern model vs. audio change point). The two distributions are multiplied pointwise, normalized to sum to unity, and a deviation value is drawn from the resulting distribution.

3.4.3 Evaluation

The proposed cut timing framework was evaluated on a user study against an automatic baseline method [76] as well as manual editing. The user study was conducted in the form of a web survey with 14 video comparison tasks. In each comparison task the user was shown a pair of videos edited from the same multi-camera video material with two of the three editing methods. For each pair of videos the user was asked to pick the video with more pleasant cut timing. The choice of methods was randomized separately for each comparison task and weighted according to the choices for previous users, so that for each comparison task all three methods were chosen roughly equal amount of times over all users. Altogether 24 users participated in the study resulting in 336 comparisons between the different editing methods. Over all users and comparison tasks the proposed method was compared 112 and 110 times to the baseline and to manual editing, respectively. The

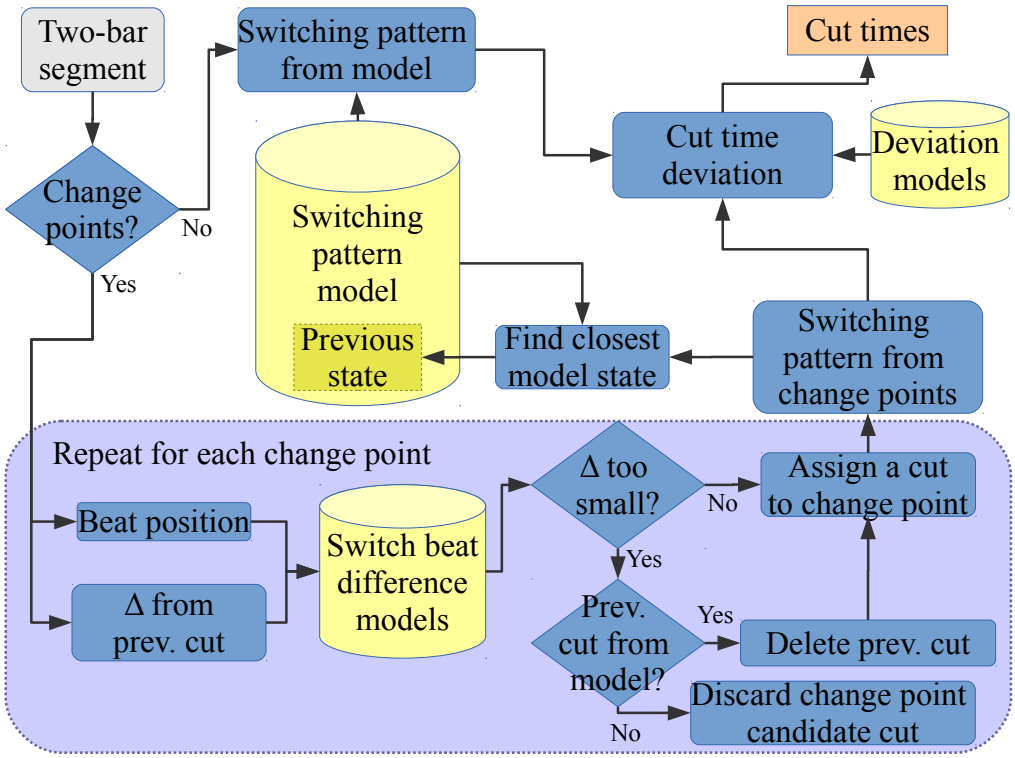

Figure 3.5: Given a two-bar segment with information about possible audio change points occurring during the segment, the cut times are assigned according to this flow diagram. Reproduced with permission from publication 4.

baseline and manual editing were compared 114 times. The study used multi-camera videos of three different concerts from the Jiku dataset [114], which contains multi-device video recordings of public performance events. Table 3.1 shows the amount and the percentage of the user study comparison wins of the method on a given row against the method on a given column. The last column shows all comparison wins of the method, and the last row all comparison losses. Figure 3.6 shows the difference of comparison wins and losses of each editing method separately for all comparison tasks. The baseline never achieves more wins than losses, and the proposed method never has the single worst win−loss ratio. Table 3.2 shows the winning percentages of different subsets of the comparison tasks, i.e., tasks with and without detected audio change points, tasks with different amounts of music structure boundaries, as well as tasks from the three different concerts of the dataset. The ranking of the three editing procedures – in terms of comparison win percentage – matches the overall ranking in all subsets, except for videos from the second concert, where the proposed method surpasses handmade editing. This event is a smaller-scale indoor concert recorded with high audio quality suiting well the audio analysis of the proposed work. All in all, compared to handmade editing the proposed method seems to take less risks in assigning the cut times, as it never has the sole worst comparison ratio in figure 3.6, whereas handmade editing has the worst ratio in three comparison tasks. More details on the evaluation can be found in publication 4.

Table 3.1: Comparison winning matrix of the three editing procedures. Reproduced with permission from publication 4.

	Proposed	Manual	Baseline	All wins
Proposed	-	53 (48.2 %)	73 (65.2 %)	126 (56.8 %)
Manual	57 (51.8 %)	-	80 (70.2 %)	137 (61.2 %)
Baseline	39 (34.8 %)	34 (29.8 %)	-	73 (32.3 %)
All losses	96 (43.2 %)	87 (38.8 %)	153 (67.7 %)	-

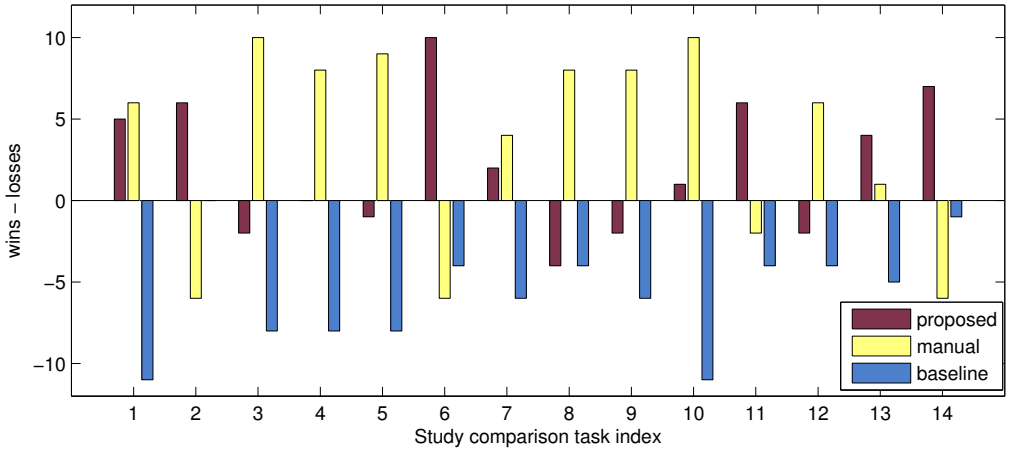

Figure 3.6: Comparison wins of each editing procedure subtracted by their comparison losses over all users for each user study comparison task. Reproduced with permission from publication 4.

Table 3.2: Relative comparison performance for different comparison task subsets. Abbreviations: PH: proposed winning over hand-made, HB: hand-made winning over baseline, BP: baseline winning over proposed, PW: total wins of proposed, HW: total wins of hand-made, BW: total wins of baseline, N: total subset comparison amount, MC + ACP: cuts from the MC model and audio change points, MC: cuts only from the MC model. Reproduced with permission from publication 4.

Subset	PH	HB	BP	PW	HW	BW	N
MC + ACP	47.3 %	74.1 %	36.4 %	55.5 %	63.7 %	31.0 %	168
MC	49.1 %	66.1 %	33.3 %	58.0 %	58.6 %	33.6 %	168
0 struct. boundaries	43.8 %	68.8 %	31.3 %	56.3 %	62.5 %	31.3 %	48
1 struct. boundary	49.3 %	68.9 %	33.8 %	57.7 %	60.0 %	32.4 %	216
2 struct. boundaries	47.8 %	75.0 %	40.0 %	54.2 %	63.8 %	32.7 %	72
Concert #1	46.2 %	78.6 %	35.9 %	55.1 %	66.7 %	28.4 %	120
Concert #2	50.0 %	62.5 %	33.3 %	58.3 %	56.3 %	35.4 %	72
Concert #3	48.9 %	66.7 %	34.7 %	57.3 %	58.9 %	34.0 %	144

4 Conclusions

This dissertation presented research on multimodal analysis in selected mobile video applications. The video medium lends itself naturally to multimodal processing as it usually incorporates both the visual and the aural stream. Mobile devices also offer a plethora of other sensors, which can be integrated with video. Multimodal fusion was applied on the recognition of everyday environments from video and audio, as well as on classification of sport type from sets of concurrent multi-device videos along with the corresponding audio and recording device motion sensor data recorded at sport events. The environment classification work considered simple global low-level visual features from video keyframes to keep the computational complexity low. This was complemented with audio event histogram based environment soundscape modeling. In this setting, training a classifier for the fusion was shown to outperform GA-optimized weighted rule-based methods. The sport type classification work compared a large collection of different fusion strategies and modality quality based adaption approaches. While different SVM classifier fusion methods gave good results on individual videos, majority fusion of crisp class predictions outperformed more complex methods in aggregating the predictions to sets of multi-device videos from a common event. All in all, multimodal analysis was clearly shown to improve the classification performance in the two tasks, which was to be expected given the rich complementary information of the used modalities. The sparse video frame sampling rate choices suit the classification of the relatively long videos in both tasks, but temporally denser visual analysis would be required for classification in finer granularity.

The experimental results support the general consensus in multimodal fusion literature that no single fusion approach dominates over different task granularities (e.g., classification of key frames vs. videos vs. sets of multi-camera videos) and data sets or applications (e.g., environment or sport type classification). However, the large variance between the performance across choices of modalities and fusion methods shows that proper optimization to a given task can make a difference between a good and an unusable multimodal analysis system. An uninformed choice of fusion components and methods can actually result in worse performance than some or even all of the components. Although different variations of learning-based fusion give good results on the two different multimodal fusion applications considered in the dissertation, a minor change in the analysis granularity of the sport type classification application results in a computationally simpler method overperforming learning-based fusion. Additionally, the granularity change also affects the optimal fusion level as the best accuracy on individual videos is achieved with early fusion, but late fusion performs better in aggregating the predictions to events consisting of multiple videos. However, drawing more general conclusions about the preference of different fusion methods and levels would require a considerably larger set of tasks and data sets.

The considered modalities have different information "bandwidths" affecting on one hand the breadth of their applicability to different tasks and on the other hand the degree of complexity in their analysis. Audio and video offer rich, widely-applicable information stemming from their relation to the corresponding prominent human senses. Audio generally offers more invariability against the spatial relation of and occlusions between the content of interest and the capturing device. The recently renewed interest towards virtual reality and 360 degree field-of-view videos may to some extent decrease the advantage of audio in omnidirectional capture and analysis. Yet, this comes at the cost of further widening the computational complexity gap between typical audio and video analysis approaches. The complexity may vary considerably also within a single modality as exemplified by the order of magnitude different processing times between the low-level spatial and spatio-temporal visual features in the sport type classification work. All in all, video analysis can relatively efficiently solve many problems unsolvable from audio and vice versa. Although signals from auxiliary sensors embedded on mobile devices typically measure a very specific quantity and thus have a much more focused scope of applicability, this information can often be obtained with a fraction of the computational complexity of even low-level content analysis of audio or video. Recording device sensor data might not be relevant to some tasks, and the data cannot be retrieved from previously recorded videos in databases if it has not been captured and saved during recording. Yet, it is easy to think of tasks, such as indoor-outdoor classification, where specific sensors (e.g., Global Positioning System (GPS) or ambient light sensor) might provide light-weight information highly complementary to an audiovisual stream. The differences in the applicability, robustness in given situation, and efficiency of the considered modalities, i.e., their complementarity, make them ideal for multimodal fusion. In all the multimodal fusion tasks, the overall best performance is achieved by including all the considered modalities.

In chapter 3, a framework was presented for modeling the timing of shot cuts of concert videos from a data set of professionally produced concert recordings. The modeling was built on top of multi-level music meter grid and audio change point analysis with specific models for cut patterns in two-bar segments, for distribution of cut differences as measured in beats, and for relative deviations of cuts from exact beat times. Models from the framework could be used for cross-modal synthesis of video shot cut times from audio. The cut timing framework can be used as part of an automatic multi-camera mashup system for creating mashup videos from live music recordings. The audio-based cut timing synthesis avoids any costly visual content analysis, yet producing acceptable cut times for multi-camera mashup videos in the experiments. The feasibility of the proposed framework and its output was confirmed with user studies, where pairs of videos were shown to the user for comparison. The pairs were formed from the same video material edited with two different methods randomly chosen between a baseline, the proposed method, and manual editing.

Cross-modal processing offers interesting opportunities for linking information in one modality to another via shared dependencies. This opens up novel applications infeasible with single modalities. Even in tasks, where unimodal analysis is possible, the multimodal angle might provide clear gains in performance, efficiency, or robustness. As an example, assigning cut timing of music videos based on the music content has evident advantages over visual-only editing: the cutting can be aligned with events from the music for creating a strong aesthetic connection between the audio and video with relatively low computational cost. The music video domain generally has strong dependencies between audio and video as the structure and events of music are often reflected in the video

structure and content (e.g., showing the source or interpretation of the audio events). As the user study evaluation results show, the concert video cut timing is a creative task feasible for automation with contemporary automatic multimedia content analysis methods. Automatic and assisted tools for improving the artistic quality in many similar creative tasks are expected to soon become as ubiquitous and as widely used as tools for automatic technical quality enhancement – such as autofocus or automatic gain control – are today.

4.1 Future work

Multimodal fusion for environment classification could naturally benefit from more intelligent content analysis of the visual frames, but more importantly proper addressing of the temporal dependencies within the videos would surely improve the results in terms of accuracy and robustness. While the classification-based fusion outperformed rule-based fusion in the experiments, the modality weight search might still be improvable by alternative optimization schemes as explicit exclusion of some of the fusion components improved the results although exclusive solutions should have been reachable also with the optimization by assigning zero weights to some components. However, more improvements would be expected from dynamic adaption between the modalities based on the content and data quality. The classification could also benefit from the complementary information of mobile device positioning systems, when available.

In the sport video classification work, the domain and data set specific assumptions should be relaxed and any hand-defined parameters properly optimized from data. Specifically, a larger set of diverse sports should be considered and, e.g., any false assumptions of typical colors of the sport venues of certain sport types should be dropped. The separation of foreground and background information should also be done more intelligently in a content-based manner, and the analysis should be adapted for cases with multiple sport types taking place concurrently, such as track and field events. The quality estimates of the different modalities could also be complemented with additional information for more robust dynamical modality weighting.

The proposed cut timing approach nearly matching the user preference of manually edited videos in the user study is encouraging. However, many potential paths exist for improving the cut timing modeling and synthesis. The modeling makes no severely limiting assumptions about the type or genre of the music (except for the time signature) and the framework can thus flexibly be used for different types of music by using a set of suitable videos as training data. However, a logic for dynamically switching between many such sub-models on the fly based on the detected genre or some other musical attribute would aid in adapting the statistical models better for the edited content. The adaption could also be done by blending between various sub-models created from different musical attributes, e.g., by weighted multiplication and normalization of the corresponding distributions, for further online adaption. Besides improving the adaption of the statistical models to the content under editing, other instantaneous cues in addition to audio change points – such as drum fills, or beat drops in electronic music – could be detected from the audio track of the edited material as triggers for cuts. In order to avoid contradicting cuts, the visual content should also be taken into account in the timing, e.g., to prevent cutting while the camera is panning. The time signature assumption should also be relaxed to broaden the applicability of the system to more complex songs.

Bibliography

[1] R. Yan and A. G. Hauptmann, "A review of text and image retrieval approaches for broadcast news video," *Information Retrieval*, vol. 10, no. 4-5, pp. 445–484, 2007.

[2] D. Brezeale and D. J. Cook, "Automatic video classification: A survey of the literature," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 3, pp. 416–430, 2008.

[3] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 797–819, Nov 2011.

[4] P. Maragos, P. Gros, A. Katsamanis, and G. Papandreou, "Cross-modal integration for performance improving in multimedia: A review," *Multimodal Processing and Interaction*, p. 1, 2008.

[5] P. K. Atrey, M. A. Hossain, A. El-Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia Syst.*, vol. 16, no. 6, pp. 345–379, 2010. [Online]. Available: http://dx.doi.org/10.1007/s00530-010-0182-0

[6] S. Shivappa, M. Trivedi, and B. Rao, "Audiovisual information fusion in human computer interfaces and intelligent environments: A survey," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1692–1715, Oct 2010.

[7] S. Essid and G. Richard, "Fusion of Multimodal Information in Music Content Analysis," in *Multimodal Music Processing*, ser. Dagstuhl Follow-Ups, M. Müller, M. Goto, and M. Schedl, Eds. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, vol. 3, pp. 37–52. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2012/3465

[8] A. Katsaggelos, S. Bahaadini, and R. Molina, "Audiovisual fusion: Challenges and new approaches," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1635–1653, Sept 2015.

[9] N. Poh, T. Bourlai, and J. Kittler, "Multimodal information fusion," *Multimodal signal processing theory and applications for human computer interaction*, p. 153, 2010.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[11] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143. [Online]. Available: http://dl.acm.org/citation.cfm?id=1643031.1643047

[12] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, Sept 2015.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[14] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *International Conference on Machine Learning (ICML)*, Bellevue, USA, June 2011.

[15] A. Jaimes and N. Sebe, "Multimodal human-computer interaction: A survey," *Comput. Vis. Image Underst.*, vol. 108, no. 1-2, pp. 116–134, Oct. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2006.10.019

[16] D. Lahat, T. Adali, and C. Jutten, "Challenges in multimodal data fusion," in *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, Sept 2014, pp. 101–105.

[17] Y. Zheng, "Methodologies for cross-domain data fusion: an overview," *Big Data, IEEE Transactions on*, vol. 1, no. 1, pp. 16–34, 2015.

[18] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[19] S. Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013.

[20] T. W. Lewis and D. M. W. Powers, "Sensor fusion weighting measures in audio-visual speech recognition," in *Proceedings of the 27th Australasian Conference on Computer Science - Volume 26*, ser. ACSC '04. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004, pp. 305–314. [Online]. Available: http://dl.acm.org/citation.cfm?id=979922.979959

[21] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *Journal of Machine Learning Research*, vol. 15, pp. 2949–2980, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14b.html

[22] C. G. M. Snoek and M. Worring, "Multimodal video indexing: A review of the state-of-the-art," *Multimedia Tools Appl.*, vol. 25, no. 1, pp. 5–35, Jan. 2005. [Online]. Available: http://dx.doi.org/10.1023/B:MTAP.0000046380.27575.a5

[23] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[24] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[25] Z.-H. Zhou, "Learning with unlabeled data and its application to image retrieval," in *PRICAI 2006: Trends in Artificial Intelligence*. Springer, 2006, pp. 5–10.

[26] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information systems*, vol. 7, no. 1, pp. 1–10, 2000.

[27] C. Sanderson and K. Paliwal, "Information fusion and person verification using speech & face information," 2002.

[28] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '05. New York, NY, USA: ACM, 2005, pp. 399–402. [Online]. Available: http://doi.acm.org/10.1145/1101149.1101236

[29] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998. [Online]. Available: http://dx.doi.org/10.1109/34.667881

[30] A. A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of Multibiometrics (International Series on Biometrics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[31] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "DeViSE: A deep visual-semantic embedding model," in *Advances In Neural Information Processing Systems, NIPS*, 2013.

[32] A. Karpathy and F. Li, "Deep visual-semantic alignments for generating image descriptions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 3128–3137. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7298932

[33] J. C. Bezdek, M. R. Pal, J. Keller, and R. Krisnapuram, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Norwell, MA, USA: Kluwer Academic Publishers, 1999.

[34] K. Sohn, W. Shang, and H. Lee, "Improved multimodal deep learning with variation of information," in *Advances in Neural Information Processing Systems*, 2014, pp. 2141–2149.

[35] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1083–1092.

[36] F. Feng, R. Li, and X. Wang, "Deep correspondence restricted boltzmann machine for cross-modal retrieval," *Neurocomputing*, vol. 154, pp. 50 – 60, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231214016841

[37] Y. Liu, X. Feng, and Z. Zhou, "Multimodal video classification with stacked contractive autoencoders," *Signal Processing*, vol. 120, pp. 761–766, 2016.

[38] W. Wang, X. Yang, B. C. Ooi, D. Zhang, and Y. Zhuang, "Effective deep learning-based multi-modal retrieval," *The VLDB Journal*, vol. 25, no. 1, pp. 79–101, 2015. [Online]. Available: http://dx.doi.org/10.1007/s00778-015-0391-4

[39] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, ser. COLT' 98. New York, NY, USA: ACM, 1998, pp. 92–100. [Online]. Available: http://doi.acm.org/10.1145/279943.279962

[40] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[41] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *International Journal of Neural Systems*, vol. 10, no. 05, pp. 365–377, 2000.

[42] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, S. Dasgupta and D. Mcallester, Eds., vol. 28, no. 3. JMLR Workshop and Conference Proceedings, May 2013, pp. 1247–1255. [Online]. Available: http://jmlr.org/proceedings/papers/v28/andrew13.pdf

[43] S. Särkkä, *Bayesian Filtering and Smoothing*. New York, NY, USA: Cambridge University Press, 2013.

[44] L. Spinello, R. Triebel, and R. Siegwart, "A trained system for multimodal perception in urban environments," in *Proceedings of the Workshop on Safe Navigation in Open and Dynamic Environment: Applications to Autonomous Vehicles (ICRA)*, 2009.

[45] N. Checka, K. Wilson, V. Rangarajan, and T. Darrell, "A probabilistic framework for multi-modal multi-person tracking," in *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, vol. 9. IEEE, 2003, pp. 100–100.

[46] K. Nickel, T. Gehrig, R. Stiefelhagen, and J. McDonough, "A joint particle filter for audio-visual speaker tracking," in *Proceedings of the 7th international conference on Multimodal interfaces*. ACM, 2005, pp. 61–68.

[47] T. Germa, F. Lerasle, N. Ouadah, and V. Cadenat, "Vision and rfid data fusion for tracking people in crowds by a mobile robot," *Computer Vision and Image Understanding*, vol. 114, no. 6, pp. 641–651, 2010.

[48] V. Kilic, M. Barnard, W. Wang, and J. Kittler, "Adaptive particle filtering approach to audio-visual tracking," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*. IEEE, 2013, pp. 1–5.

[49] M. Gabbouj, "Novel dsp tools in cbir muvis framework," in *Proceedings of the Second Annual IEEE Benelux/DSP Valley Signal Processing Symposium, SPS-DARTS 2006, Metropolis, Antwerp, Belgium, 28-29 March 2006*, 2006.

[50] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Audio context recognition using audio event histograms," in *Signal Processing Conference, 2010 18th European*. IEEE, 2010, pp. 1272–1276.

[51] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[52] J. C. Platt, "Advances in kernel methods," B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208. [Online]. Available: http://dl.acm.org/citation.cfm?id=299094.299105

[53] J. C. Platt, "Probabilities for SV machines," in *Advances in Large Margin Classifiers*. MIT Press, March 1999, pp. 61–74. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=69187

[54] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. Mach. Learn. Res.*, vol. 5, pp. 975–1005, Dec. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1005332.1016791

[55] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 3, no. 6, pp. 610 –621, nov. 1973.

[56] M. Partio, B. Cramariuc, and M. Gabbouj, "An ordinal co-occurrence matrix framework for texture retrieval," *J. Image Video Process.*, vol. 2007, no. 1, pp. 1–1, 2007.

[57] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 703 –715, jun. 2001.

[58] M. Wall, "GAlib: A C++ library of genetic algorithm components," Mechanical Engineering Department, Massachusetts Institute of Technology, Tech. Rep., 1996.

[59] P. Jourlin, "Word-dependent acoustic-labial weights in HMM-based speech recognition," in *Audio-Visual Speech Processing: Computational & Cognitive Science Approaches*, 1997, pp. 69–72.

[60] J. Wang, C. Xu, and E. Chng, "Automatic sports video genre classification using pseudo-2D-HMM," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 4, 2006, pp. 778–781.

[61] F. Cricri, *Multimodal analysis of mobile videos*, ser. Tampere University of Techology. Publication. Tampere University of Technology, 2014, awarding institution:Tampere University of Technology.

[62] I. Laptev and T. Lindeberg, "Space-time interest points," in *IN ICCV*, 2003, pp. 432–439.

[63] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Department of Computer Science, National Taiwan University, Tech. Rep., 2003. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers.html

[64] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in SVMs," in *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, 2002, pp. 553–560. [Online]. Available: http://papers.nips.cc/paper/2156-adaptive-scaling-for-feature-selection-in-svms

[65] O. Chapelle and S. S. Keerthi, "Multi-class feature selection with support vector machines," in *Proceedings of the American statistical association*, 2008.

[66] G. Forman, M. Scholz, and S. Rajaram, "Feature shaping for linear SVM classifiers," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, 2009, pp. 299–308. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557057

[67] Q.-C. Wang, W. W. Ng, P. P. Chan, and D. S. Yeung, "Feature weighting based on l-gem," in *2010 International Conference on Machine Learning and Cybernetics*, 2010.

[68] S. Kiranyaz, S. Uhlmann, J. Pulkkinen, M. Gabbouj, and T. Ince, "Collective network of evolutionary binary classifiers for content-based image retrieval," in *Evolving and Adaptive Intelligent Systems (EAIS), 2011 IEEE Workshop on.* IEEE, 2011, pp. 147–154.

[69] J. M. Zacks, T. S. Braver, M. A. Sheridan, D. I. Donaldson, A. Z. Snyder, J. M. Ollinger, R. L. Buckner, and M. E. Raichle, "Human brain activity time-locked to perceptual event boundaries," *Nature neuroscience*, vol. 4, no. 6, pp. 651–655, 2001.

[70] O. Gillet, S. Essid, and G. Richard, "On the correlation of automatic audio and visual segmentations of music videos," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 3, pp. 347–355, March 2007.

[71] H. Zettl, *Sight, Sound, Motion: Applied Media Aesthetics*, ser. Wadsworth series in broadcast and production. Wadsworth Cengage Learning, 2011.

[72] J. Wang, E. Chng, C. Xu, H. Lu, and Q. Tian, "Generation of personalized music sports video using multimodal cues," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 576–588, 2007.

[73] U. Naci, "Multimedia content analysis, indexing and summarization: A perspective on real-life uses cases," PhD thesis, Technische Universiteit Delft, 2010. [Online]. Available: http://repository.tudelft.nl/view/ir/uuid:93784cba-42d9-4d94-a430-0aacfa01bf28/

[74] L. Kennedy and M. Naaman, "Less talk, more rock: Automated organization of community-contributed collections of concert videos," in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 311–320. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526752

[75] X. Shao, C. Xu, N. C. Maddage, Q. Tian, M. S. Kankanhalli, and J. S. Jin, "Automatic summarization of music videos," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 2, pp. 127–148, May 2006.

[76] X.-S. Hua, L. Lu, and H.-J. Zhang, "Automatic music video generation based on temporal pattern analysis," in *Proceedings of the 12th annual ACM international conference on Multimedia*, ser. MULTIMEDIA '04. New York, NY, USA: ACM, 2004, pp. 472–475.

[77] X.-S. Hua, L. Lu, and H.-J. Zhang, "Optimization-based automated home video editing system," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 5, pp. 572–583, 2004.

[78] J.-C. Yoon, I.-K. Lee, and S. Byun, "Automated music video generation using multi-level feature-based segmentation," *Multimedia Tools Appl.*, vol. 41, no. 2, pp. 197–214, Jan. 2009.

[79] N. Nitta and N. Babaguchi, "Example-based video remixing," *Multimedia Tools Appl.*, vol. 51, no. 2, pp. 649–673, Jan. 2011.

[80] C. Liao, P. P. Wang, and Y. Zhang, "Mining association patterns between music and video clips in professional MTV," in *Proceedings of the 15th International Multimedia Modeling Conference on Advances in Multimedia Modeling*, ser. MMM '09. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 401–412. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-92892-8_41

[81] C. G. M. Snoek, M. Worring, A. Smeulders, and B. Freiburg, "The role of visual content and style for concert video indexing," in *Multimedia and Expo, 2007 IEEE International Conference on*, 2007, pp. 252–255.

[82] L.-Y. Duan, J. S. Jin, Q. Tian, and C.-S. Xu, "Nonparametric motion characterization for robust classification of camera motion patterns," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 323–340, April 2006.

[83] M. A. Hasan, M. Xu, X. He, and C. Xu, "Camhid: Camera motion histogram descriptor and its application to cinematographic shot classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1682–1695, Oct 2014.

[84] F. Cricri, K. Dabov, M. J. Roininen, S. Mate, I. D. D. Curcio, and M. Gabbouj, "Multimodal semantics extraction from user-generated videos," *Advances in Multimedia*, vol. 2012, p. 1, 2012.

[85] H. L. Wang and L. F. Cheong, "Taxonomy of directing semantics for film shot classification." *IEEE Trans. Circuits Syst. Video Techn.*, vol. 19, no. 10, pp. 1529–1542, 2009. [Online]. Available: http://dblp.uni-trier.de/db/journals/tcsv/tcsv19.html#WangC09

[86] P. Shrestha, P. H. de With, H. Weda, M. Barbieri, and E. H. Aarts, "Automatic mashup generation from multiple-camera concert recordings," in *Proceedings of the international conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 541–550.

[87] G. Schofield, T. Bartindale, and P. Wright, "Bootlegger: Turning fans into film crew," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 767–776. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702229

[88] S. Wilk, S. Kopf, and W. Effelsberg, "Video composition by the crowd: A system to compose user-generated videos in near real-time," in *Proceedings of the 6th ACM Multimedia Systems Conference*, ser. MMSys '15. New York, NY, USA: ACM, 2015, pp. 13–24. [Online]. Available: http://doi.acm.org/10.1145/2713168.2713178

[89] J.-C. Wang, M.-C. Yen, Y.-H. Yang, and H.-M. Wang, "Automatic set list identification and song segmentation for full-length concert videos." in *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, 2014, pp. 239–244.

[90] G. Sargent, P. Hanna, and H. Nicolas, "Segmentation of music video streams in music pieces through audio-visual analysis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 724–728.

[91] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of trecvid activity," *Comput. Vis. Image Underst.*, vol. 114, no. 4, pp. 411–418, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2009.03.011

[92] J. Foote, M. Cooper, and A. Girgensohn, "Creating music videos using automatic media analysis," in *Proceedings of the tenth ACM international conference on Multimedia*, ser. MULTIMEDIA '02. New York, NY, USA: ACM, 2002, pp. 553–560.

[93] Y. Y. Xiang and M. S. Kankanhalli, "Automated aesthetic enhancement of videos," in *Proceedings of the International Conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 281–290. [Online]. Available: http://doi.acm.org/10.1145/1873951.1873991

[94] X. Wu, Y. Qiao, X. Wang, and X. Tang, "Cross matching of music and image," in *Proceedings of the 20th ACM International Conference on Multimedia*, ser. MM '12. New York, NY, USA: ACM, 2012, pp. 837–840. [Online]. Available: http://doi.acm.org/10.1145/2393347.2396325

[95] P. Mulhem, M. Kankanhalli, J. Yi, and H. Hassan, "Pivot vector space approach for audio-video mixing," *MultiMedia, IEEE*, vol. 10, no. 2, pp. 28–40, 2003.

[96] J.-C. Wang, Y.-H. Yang, I.-H. Jhuo, Y.-Y. Lin, and H.-M. Wang, "The acousticvisual emotion Guassians model for automatic generation of music video," in *Proceedings of the 20th ACM International Conference on Multimedia*, ser. MM '12. New York, NY, USA: ACM, 2012, pp. 1379–1380. [Online]. Available: http://doi.acm.org/10.1145/2393347.2396494

[97] R. Cai, L. Zhang, F. Jing, W. Lai, and W. Ma, "Automated music video generation using WEB image resource," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007*, 2007, pp. 737–740. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2007.366341

[98] D. A. Shamma, B. Pardo, and K. J. Hammond, "MusicStory: A personalized music video creator," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '05. New York, NY, USA: ACM, 2005, pp. 563–566. [Online]. Available: http://doi.acm.org/10.1145/1101149.1101278

[99] S. Xu, T. Jin, and F. Lau, "Automatic generation of music slide show using personal photos," in *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, Dec 2008, pp. 214–219.

[100] X. Wu, B. Xu, Y. Qiao, and X. Tang, "Automatic music video generation: Cross matching of music and image," in *Proceedings of the 20th ACM International Conference on Multimedia*, ser. MM '12. New York, NY, USA: ACM, 2012, pp. 1381–1382. [Online]. Available: http://doi.acm.org/10.1145/2393347.2396495

[101] T. Chen, C.-W. Chen, P. Popp, and B. Coover, "Visual rhythm detection and its applications in interactive multimedia," *MultiMedia, IEEE*, vol. 18, no. 1, pp. 88–95, 2011.

[102] W. T. Chu and S. Y. Tsai, "Rhythm of motion extraction and rhythm-based cross-media alignment for dance videos," *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 129–141, Feb 2012.

[103] C. Panagiotakis, A. Holzapfel, D. Michel, and A. A. Argyros, *Advances in Visual Computing: 9th International Symposium, ISVC 2013, Rethymnon, Crete, Greece, July 29-31, 2013. Proceedings, Part II.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ch. Beat Synchronous Dance Animation Based on Visual Analysis of Human Motion and Audio Analysis of Music Tempo, pp. 118–127. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-41939-3_12

[104] C. Ho, W. T. Tsai, K. S. Lin, and H. H. Chen, "Extraction and alignment evaluation of motion beats for street dance," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 2429–2433.

[105] T. Nakano, S. Murofushi, M. Goto, and S. Morishima, "DanceReProducer: An automatic mashup music video generation system by reusing dance video clips on the web," in *Proc. of the 8th Sound and Music Computing Conference (SMC 2011)*, 2011, pp. 183–189.

[106] H. Ohya and S. Morishima, "Automatic mash up music video generation system by remixing existing video content," in *Culture and Computing (Culture Computing), 2013 International Conference on*, Sept 2013, pp. 157–158.

[107] Z. Liao, Y. Yu, B. Gong, and L. Cheng, "Audeosynth: Music-driven video montage," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 68:1–68:10, Jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2766966

[108] M. K. Saini, R. Gadde, S. Yan, and W. T. Ooi, "MoViMash: online mobile video mashup," in *Proceedings of the 20th ACM international conference on Multimedia*, ser. MM '12. New York, NY, USA: ACM, 2012, pp. 139–148.

[109] I. Arev, H. S. Park, Y. Sheikh, J. Hodgins, and A. Shamir, "Automatic editing of footage from multiple social cameras," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 81:1–81:11, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2601097.2601198

[110] S. Bano and A. Cavallaro, "ViComp: composition of user-generated videos," *Multimedia tools and applications*, pp. 1–24, 2015.

[111] Y. Wu, T. Mei, Y. Q. Xu, N. Yu, and S. Li, "Movieup: Automatic mobile video mashup," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 1941–1954, Dec 2015.

[112] A. Eronen and A. Klapuri, "Music tempo estimation with k-NN regression," *IEEE Audio, Speech, Language Process.*, vol. 18, no. 1, pp. 50–57, 2010.

[113] D. P. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[114] M. Saini, S. P. Venkatagiri, W. T. Ooi, and M. C. Chan, "The Jiku mobile video dataset," in *Proceedings of the 4th ACM Multimedia Systems Conference*, ser. MMSys '13. New York, NY, USA: ACM, 2013, pp. 108–113. [Online]. Available: http://doi.acm.org/10.1145/2483977.2483990

Errata and Clarifications for the Publications

- In publication 1 the part stating " ... and $\check{\mathbf{l}}_{i(j)}$ the likelihood vector of the ith expert excluding the jth likelihood." should be " ... and $\check{\mathbf{l}}_{i(j)}$ the multimodal likelihood vector of the jth class excluding the ith expert."
- In publication 3 the sentence "(ii) results from the first stage for different classes are fused by majority voting in the "Stage 2 Fuser"." should be corrected as "(ii) results from the first stage for different classes are fused by picking the most confident single-class prediction in the "Stage 2 Fuser"."
- In publication 4 the sentence "The LDA is performed between the classes "first beat of a group of two bars" and "other beat"." should be appended with " ... and produces a two-bar downbeat likelihood score sequence s_{2b}.". Correspondingly, "The different signals s_{db}, s_{cc}, s_{no}, and s_{cp} ... " should be corrected as "The different signals s_{2b}, s_{cc}, s_{no}, and s_{cp} ... ". This clarification explicitly states the difference of the LDA classification in the bar and two-bar cases.

Publications

Publication I

Mikko Roininen, Esin Guldogan, Moncef Gabbouj, "Audiovisual video context recognition using SVM and genetic algorithm fusion rule weighting," *in Content-Based Multimedia Indexing (CBMI), 2011 9th International Workshop on*, pp. 175 – 180 Jun. 2011.

Publication II

Francesco Cricri, Mikko Roininen, Sujeet Mate, Jussi Leppänen, Igor D. D. Curcio, Moncef Gabbouj, "Multi-sensor fusion for sport genre classification of user generated mobile videos," *in Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pp.1-6, 15-19 July 2013.

Publication III

Francesco Cricri, Mikko Roininen, Jussi Leppänen, Sujeet Mate, Igor D. D. Curcio, Stefan Uhlmann, Moncef Gabbouj, "Sport Type Classification of Mobile Videos," *in Multimedia, IEEE Transactions on*, vol.16, no.4, pp.917-932, June 2014.

Publication IV

Mikko Roininen, Jussi Leppänen, Igor D. D. Curcio, Moncef Gabbouj, "Modeling the timing of cuts in automatic editing of concert videos," *in Multimedia Tools and Applications*, 2016, pp 1 – 25, DOI 10.1007/s11042-016-3304-7. With permission of Springer.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3845-2
ISSN 1459-2045