Karlsruher Institut für Technologie

Contextual Person Identification in Multimedia Data

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)
genehmigte

Dissertation
von

Dipl.-Inform. Martin Bäuml

aus Erlangen

Tag der mündlichen Prüfung:	18. November 2014
Hauptreferent:	Prof. Dr. Rainer Stiefelhagen Karlsruher Institut für Technologie
Korreferent:	Prof. Dr. Gerhard Rigoll Technische Universität München

Abstract

Automatic tracking and identification of faces and persons are essential tasks in many video analysis systems, for example to automatically generate meta data or as basis for higher level applications. In many cases, identification is based on a single modality such as faces. In this work, we propose methods to improve person identification by integration of multiple cues including multiple modalities and contextual information.

We motivate and evaluate our proposed methods in the context of multimedia data, specifically TV series. Despite its usually high resolution, multimedia data presents many challenges. For example, camera views change constantly at shot boundaries, the camera position is generally unknown and image conditions and poses of faces and poses can change rapidly due to the underlying plot. Since we make only few assumptions about the underlying data, our methods are applicable to other domains as well, for example in the area of safety and security.

Before we can identify a face, it has to be localized in the image first. In videos we can further associate localizations over time to consecutive face tracks. Face tracks can then be identified jointly and errors in single frames (*e.g.*, due to noise in the data or imprecise localization) can be mitigated, improving overall identification accuracy. In this work, we propose a detector-based face tracking approach based on a large bank of detectors which cover a range of head poses. We integrate the detectors in a particle filter such that these can be used efficiently, *i.e.* only one detector out of 49 is evaluated for each particle.

We evaluate our approach on a data set of two TV series, which we annotated with ground truth face positions. The data set contains over 100 000 annotated faces and is one of the largest public data sets available for the evaluation of face tracking. With our proposed tracking approach we achieve an improvement of 0.15 in Multiple Object

Tracking Accuracy over a frontal-face-only tracker. The miss rate (faces that are not found) reduces by an absolute 10-15%, while the false positive rate only increases by an absolute 5%. Consequently, the number of underlying tracks for which at least one frame is found by the tracker also increases: the track recall improves by an absolute 5-15% while maintaining a high track precision.

While previous work in multimedia data focused on the recognition of faces, we extend the problem to recognize persons, regardless of the visibility of the face. We employ a person tracker in addition to face tracking approach as described above to localize persons where the face is not visible. We approach the identification of persons from two perspectives. On the one hand, we propose a learning approach that integrates additional contextual information from videos to improve the learned face models. For learning a face model, we use both labeled data, for which we generate labels automatically by matching subtitles and transcripts, and unlabeled data as well as constraints between face tracks. The different sources of information are combined in a joint loss function. On the other hand, we integrate additional modalities, *e.g.* the general appearance of a person or his/her expected presence on the screen, in a global fusion framework.

Due to the integration of additional information in the learning of face models, we achieve an improvement in track identification accuracy of around 2% on average. By fusing with additional modalities, the accuracy on *face* tracks improves by another 4% on average. For *person* tracks, the accuracy improves due to the integration of multiple modalities by an absolute of over 11%, since this allows to also identify persons correctly for which the face is not visible or the face tracker could not find the face.

Finally, we propose a family of kernels which operate on tracks instead of single frames/features. These kernels are applicable in many kernel-based learning approaches. Due to the usage of tracks instead of frames as underlying basic units for learning, the computational requirement for the optimization of the loss function is reduced by a factor proportional to the square of the average track length.

The reduction in training time is especially of importance, when manual feedback is considered during learning. Due to the reduced complexity, we can train new face models and infer identities for all tracks within seconds. Therefore, the influence of the manual labels can be increased. When correcting misclassified face tracks, less than half of the tracks have to be corrected when we train and evaluate new models during feedback. This is equivalent reducing the required manual label time by more than 50%.

Kurzzusammenfassung

Das automatische Nachverfolgen und Identifizieren von Gesichtern und Personen ist Grundlage in vielen automatischen Videoanalysesystemen, z.B. zur automatischen Generierung von Metadaten oder als Basis für darauf aufbauende weiterführende Anwendungen. In vielen Fällen wird eine Personenidentifikation nur auf einer Modalität basierend durchgeführt, z.B. Gesichtern. In dieser Arbeit untersuchen wir Methoden, durch die die Identifikation von Personen durch Hinzunahme von weiteren Modalitäten und anderen (Kontext-)Informationen verbessert werden kann.

Wir motivieren und evaluieren unsere Verfahren am Beispiel von Multimediadaten (TV Serien). Trotz der oft hohen Auflösung sind Multimediadaten herausfordernd, da sich Blickwinkel aufgrund der Kameraführung ständig ändern, im Allgemeinen keine Kalibrierung der Kamera bekannt ist und sich durch die unterliegende Handlung die Aufnahmebedingungen und Ansichten von Personen und Gesichtern in schneller Abfolge ändern können. Da wir nur wenige Annahmen zu den unterliegenden Daten treffen, sind die entwickelten Verfahren weitestgehend ebenso in anderen Domänen, z.B. im Sicherheitsbereich, einsetzbar.

Bevor ein Gesicht identifiziert werden kann, muss es zunächst im Bild lokalisiert werden. In Videos kann zusätzlich ein zeitlicher Zusammenhang zwischen einzelnen Gesichtern hergestellt werden. Dies erleichtert eine spätere Identifizierung, da zusammenhängende Gesichtstracks gemeinsam identifiziert werden können und Fehler in einzelnen Frames (z.B. durch Rauschen in den Daten oder aufgrund einer ungenauen Lokalisierung) ausgeglichen werden können. Aufgrund obengenannter Eigenschaften ist in realistischen Daten allerdings schon die Lokalisierung und Nachverfolgung (Tracking) von Gesichtern in Videosequenzen ein herausforderndes Problem. In den letzten Jahren haben sich diskriminative Objektdetektoren zur Lokalisierung von Objekten in Bildern durchgesetzt und – darauf aufbauend – detektor-basierte Tracking-Ansätze für Videos. Um unterschiedliche Posen eines Gesichts abzudecken, werden oft viele, posen-spezifische Detektoren eingesetzt. Durch die höhere Anzahl an Detektoren steigt allerdings auch die benötigte Laufzeit.

In dieser Arbeit stellen wir einen detektor-basierten Tracking-Ansatz für Gesichter vor, der zwar auf einer großen Anzahl von Detektoren beruht, durch die Integration in ein Partikelfilter diese Detektoren aber Laufzeit-effizient einsetzt. So wird anstatt unserer kompletten Detektorbank von 49 Detektoren nur jeweils ein Detektor pro Partikel ausgewertet.

Wir evaluieren unseren Tracking-Ansatz auf einem Datensatz von zwei TV Serien, die wir zu diesem Zweck mit Grundwahrheit-Lokalisierungen annotiert haben. Der Datensatz enthält über 100 000 annotierte Gesichter und ist einer der größten veröffentlichten Datensätze zur Evaluation von Gesichtstracking. Im Vergleich zu einem Tracker, der nur auf einem frontalen Detektor basiert, erzielen wir eine durchschnittliche Verbesserung der Multiple Object Tracking Accuracy um 0.15. Vor allem die Miss-Rate (nicht gefundene Gesichter) sinkt je nach Video durch die Verwendung von mehr Detektoren um 10-15 Prozentpunkte, während die Falsch-Positiv-Rate nur um bis zu ca. 5 Prozentpunkte zunimmt. Dadurch erhöht sich auch die Anzahl der unterliegenden Gesichtssequenzen, von denen zumindest ein Frame gefunden wurde: Der Track Recall steigt konsistent um 5-15 Prozentpunkte, während die Track Precision konstant bleibt oder ebenfalls ansteigt.

Während vorherige Arbeiten in Multimediadaten sich rein auf die Identifikation von Gesichtern beschränkten, erweitern wir das Problem auf die Identifikation von Personen, auch wenn kein Gesicht sichtbar ist. Dazu verwenden wir zusätzlich zum oben beschriebenen Trackingverfahren einen Personentracker, der Personeninstanzen unabhängig von Gesichtern lokalisiert.

Die Identifizierung von Personen untersuchen wir aus zwei unterschiedlichen Perspektiven. Zum einen schlagen wir ein Lernverfahren vor, das zusätzliche Informationen aus gegebenen Videos verwenden kann, um die zu lernenden Gesichtsmodelle zu verbessern. Dabei verwenden wir sowohl gelabelte Daten, für die wir die Labels aus Untertiteln und Drehbuch für einen Teil der Tracks automatisch generieren, als auch ungelabelte Daten und Einschränkungen von Paaren von Tracks. Die unterschiedlichen Komponenten werden in einer gemeinsamen Kostenfunktion zusammengefasst und gemeinsam optimiert. Zum anderen integrieren wir zusätzliche Modalitäten in einem globalen

Fusions-Framework, z.B. das komplette Erscheinungsbild einer Person (bzw. ihrer Kleidung) oder die erwartete Präsenz einer Person im Bild, wenn sie gerade spricht.

Durch die Integration von zusätzlichen Information in das Lernen von Gesichtsmodellen erreichen wir eine Steigerung der Trackerkennungsrate um durchschnittlich ca. 2 Prozentpunkte. Durch die Fusion mit weiteren Modalitäten verbessert sich die Erkennungsleistung auf *Gesichts*tracks um durchschnittlich weitere 4 Prozentpunkte. Für *Personen*tracks steigert die Hinzunahme von weiteren Modalitäten die Erkennungsleistung um über 11 Prozentpunkte, da nun auch Personen richtig erkannt werden können, deren Gesicht nicht sichtbar ist oder vom Tracker nicht gefunden wurde.

Schließlich schlagen wir eine Familie von Kerneln vor, die auf Tracks anstatt einzelnen Frames operieren. Diese sind für viele Kernel-basierten Lernverfahren anwendbar. Durch die Verwendung von Tracks anstatt Frames als unterliegende Einheiten zum Lernen reduziert sich der Rechenaufwand der Optimierung der Kostenfunktion um einen Faktor proportional zum Quadrat der durchschnittlichen Tracklänge.

Die Reduktion der benötigten Rechenzeit ist insbesondere dann von Vorteil, wenn manuelles Feedback in das Lernen mit einfließen soll. Durch die Vereinfachung des Verfahrens können wir Gesichtsmodelle in wenigen Sekunden neu trainieren und auf allen Tracks evaluieren. Dadurch kann der Einfluss von manuellen Labels gesteigert werden. Bei der Korrektur von fehlerhaft erkannten Gesichtstracks muss so nur noch für weniger als die Hälfte der Tracks manuelles Feedback gegeben werden, gleichbedeutend mit einer Einsparung der Labelzeit von über 50%.

Contents

Chapter 1

Introduction

Automatic person identification in images and videos is an extensive and challenging problem. Person identities provide useful meta data for both personal and professional applications, for example in digital photo albums, media databases for human-computer interaction or security applications. They also serve as a foundation and building block for many higher level video analysis tasks such as textual image description, interaction analysis or video summarization. Ultimately, person identities are a key information to understanding the visual content of images and videos, to make it searchable and useful to others.

Large amounts of visual data are generated in different domains and from different data sources independently, with massive growth rates. With the rise of smart phones, it has never been easier to take pictures or record personal videos and it is estimated that about 10% of all pictures taken since the invention of the camera have been taken in the last 12 months (Good, 2011). Professional video sources include multimedia data (broadcast archives, TV series and movies) and surveillance camera networks. Indexing the information contained in the sheer volume of the data requires an automated solution.

In the past, the focus for automatic person identification has been placed by and large on one modality at a time, which in many cases was faces. However, person identification is an inherently multimodal problem and humans usually rely on more than just the face for inferring the identity of a person. Often, the face of a person is partially occluded, not visible at all, or not distinctive enough to make an informed decision on the identity, for example because of bad lighting conditions. “Soft biometrics” such as gender, general appearance, distinctive symbols on clothing, hair color and style, accessories, gait patterns or voice can play a great deal in telling people apart. In addition, soft and hard constraints on person identities arise naturally from images and videos. As a hard constraint for example, two persons in the same video frame should not be identified as the same person. A soft constraint can arise from observing groups of people and making assumptions about them staying a group. In multimedia data, subtitles, transcripts, overlaid text and speech provide constructive cues on person identities. Consequently, automatic person identification can benefit from using such data to make more informed and thus (hopefully) better decisions.

Automatic naming of persons in multimedia data has received increasing attention in the last years. As a direct commercial application, video streaming providers have very recently started to offer information on cast and characters for TV series and movies during playback[1,2,3]. Also in multimedia data, person identities serve as basis for higher level applications, for example to search large media data bases for specific persons. Multimedia data can usually be considered to be taken "in-the-wild", *i.e.* in varying and diverse conditions, different poses, with partial occlusions and generally without cooperation simplifying automatic identification. In long running TV series one can also observe the effects of aging, different hair styles, weight loss and gain, adding to the challenges of reliably identifying persons in such data.

In this thesis, we address as main application the naming of characters in TV series and movies and make use of special characteristics of such data. To this end, we take into account the whole pipeline of person identification and perform fully automatic end-to-end person identification. We consider the following parts and underlying problems of the pipeline in more detail.

The localization of visual elements for recognition is an underlying problem for most of the above-mentioned modalities. If a face or person is not detected in an image, automatic recognition cannot be expected. Therefore, robust localization is a necessary prerequisite and high recall and precision are key to applicability in realistic scenarios.

A second central question is how to incorporate the different modalities and contextual sources. This can be done initially while learning models for identification, *e.g.* by generating training data automatically from some of the accompanying data such as transcripts and subtitles. Unlabeled data and constraints can provide additional information to

[1] Hulu Face Match: http://www.hulu.com/labs/tagging
[2] Amazon/IMDB X-Ray for movies: http://www.imdb.com/x-ray/
[3] Actor info cards for Google Play Movies & TV

train better models. Similarly, a fusion of multiple complementary cues, *e.g.* clothing appearance, is likely to be beneficial for identification of previously unseen test data.

Finally, when learning from large amounts of data, the training set is often more constrained by the available memory than the available data. When using weak label sources such as subtitles and transcripts, in order to correct remaining errors after identification, informed human labels are required. In order to minimize the necessary human input, person models can be updated iteratively during the labeling process and thus more tracks be automatically identified correctly. A reduction in the number of elements to classify, *e.g.* tracks instead of individual features, can present a solution to both challenges.

While the application focus of this thesis lies on multimedia data, the main ideas and contributions can also be applied to other domains. For example, the presented face tracker has originally been developed in a surveillance setting (Bäuml et al., 2010b). Learning identities can similarly exploit contextual cues from a camera network (Bäuml et al., 2012), and other modalities which are not explicitly explored in this thesis, can be integrated in the proposed fusion scheme.

1.1. Overview over related literature

Early work on face identification in the "wild" was performed in news images (Berg et al., 2004). To deal with the large amount of data, the idea of leveraging the captions as weak labels and constraints on the identities of the persons depicted in the image was explored. Labeling images from captions was continued with more advanced features and different learning methods (Guillaumin et al., 2010, 2012). Similar ideas were explored in broadcast news videos (Khoury et al., 2013; Song et al., 2004; Yang et al., 2005). No direct textual captions are available here, but transcripts (Song et al., 2004; Yang et al., 2005) or Optical Character Recognition of the overlaid text (Poignant et al., 2012; Yang et al., 2005) provide equivalent information. In the end, the usage of this out-of-band information allows to identify faces in large corpora without human intervention. An interesting aspect of these approaches is that the employed labels are ambiguous, *i.e.* usually there cannot be made a one-to-one correspondence between a name in the caption and a face in the image. Learning the person model inherently has to determine the relationship between labels and possible faces during training.

To avoid manual labeling of faces in TV series for training person models, Everingham et al. (2006) proposed an automatic method to obtain weak labels for some track identities by aligning subtitles to transcripts and automatic speaker detection. Since speaker detection is a difficult problem by itself, these labels are typically noisy and incomplete (*i.e.*, usually only about 20-30% of the tracks can be assigned a name, with about 80-90% accuracy). Modifications to the original speaker detection procedure by Everingham et al. (2006) have been proposed (Köstinger et al., 2011). As in broadcast news, these labels can be regarded as partial and ambiguous while learning person models, addressed by modeling the learning problem as a multiple instance learning problem (Köstinger et al., 2011) or by using a specialized convex loss function (Cour et al., 2011). When transcripts are not available, references to names in the spoken text can be leveraged (Cour et al., 2010), although they provide much weaker hints on the identity of the faces in the current shot. Work has also been conducted to align scripts to TV series and movies when subtitles are not available (Sankar et al., 2009). However, this requires manual samples for identification and thus cannot be used directly for identification itself.

In the context of multimedia data the problems of retrieval (Fischer et al., 2010; Sivic et al., 2005) and clustering (Arandjelovic and Cipolla, 2006; Ramanan et al., 2007a; Yamamoto et al., 2010) have been similarly addressed. Feature length movies were used originally as easily obtainable data source for face recognition experiments (Fitzgibbon and Zisserman, 2003). Recognizing the potential, Ramanan et al. (2007a) stated the explicit goal of obtaining a data set for further face analysis experiments, using data from different seasons of a TV series to cover changes in age, hair style and weight. However, the data set was never widely used. Equivalently, face recognition and person identification was extended to other "wild" domains such as generic web videos (Sargin and Aradhye, 2009; Zhao and Yagnik, 2008), personal photo albums (Gallagher and Chen, 2008a, 2007; Lin et al., 2010; Zhang et al., 2013) and movie trailers (Ortiz et al., 2013).

The main focus of naming characters lies on naming *face* tracks (Everingham et al., 2006; Köstinger et al., 2011; Ortiz et al., 2013; Ramanan et al., 2007a; Sivic et al., 2009). Due to close up shots, face detection and tracking is an easier problem in such data than person detection. In addition, faces are usually assumed to remain very similar in appearance throughout a movie or TV series, while a character's clothing might change frequently between scenes. Clothing has been used as an auxiliary cue for naming faces, but the localization is based on faces, *e.g.*, the clothing descriptor is extracted from a bounding box beneath the face (Anguelov et al., 2007; Everingham et al., 2006; Ramanan et al., 2007a). Consequently, the performance of the face tracker is important, as non-localized faces are not further considered in the later stages of the identification pipeline. Different kinds of region and point-based trackers have been proposed to track over frames in which the underlying face detector could not find a face (Everingham et al., 2006; Sivic et al., 2005). In order to increase the track recall of their tracker, Sivic et al. (2009) used both a frontal and profile face detector as underlying face model. Ramanan et al. (2007a) continued tracking using person-specific part models (*i.e.*, of hair, face and torso) but restricted the continuation to frames neighboring a face detection. In the same spirit as Sivic et al. (2009), we extended our face tracker to track multiple poses with the goal to locate more faces than a frontal-only tracker (see Chapter 2). We further extended the problem to *person* tracks obtained by an independent person tracker to further increase coverage. Using independent person tracks, we are able to identify persons even when no face can be detected (see Chapter 4 for details).

Many different descriptors have been employed for face recognition. In the context of multimedia data alone, there is a wide range of explored descriptors: simple pixel-based (Everingham et al., 2006), Histograms of Oriented Gradients (HOG) (Sivic et al., 2009), local descriptors (Cinbis et al., 2011; Khoury et al., 2013), local block-based Discrete Cosine Transform (DCT) (Fischer et al., 2010), Local Binary Patterns (LBP) (Ortiz et al., 2013), Gabor-response histograms (Ortiz et al., 2013) or SIFT-based Fisher vectors (FV^2) (Parkhi et al., 2014). However, as discussed before, person identification is an inherently multi-modal problem. For example, performing speaker and face recognition together is beneficial in both directions: audio can be used to verify face recognition results and faces can help to identify the speaker (Li et al., 2001). The association between speaker and face is often performed in a greedy manner (Gianni et al., 2007; Sargin and Aradhye, 2009). For personal photo albums, Markov random fields have been employed for fusing face and clothing cues (Anguelov et al., 2007), people, events and locations (Lin et al., 2010) or faces, human attributes, clothing and co-occurrence (Zhang et al., 2013). Clothing is a strong feature and can be described quite detailed (Yamaguchi et al., 2012), however, such approaches are computationally too expensive for applicability in large-scale video data bases. Typically, approaches resort to color and texture histograms with different degree of segmentation of the clothing region (Anguelov et al., 2007; Gallagher and Chen, 2008a; Ramanan et al., 2007b; Weber et al., 2011).

Other cues which can be incorporated in the identification procedure include gender (Cour et al., 2011) and character co-occurrence (Cour et al., 2009; Sang and Xu, 2012; Zhang et al., 2009). The latter analyzes which characters are likely to appear together (or not) and as such influences character naming decisions. One more example, where specific elements of multimedia data are exploited are the usage of shot threads (Cour et al., 2008; Yamamoto et al., 2010). Due to the way movies and TV series are edited, there are often sequences which switch back and forth between two camera views, especially during conversations. Persons often do not move between these camera switches and thus the identities of the corresponding tracks can be linked together. In Chapter 4, we incorporate multiple cues, including clothing without faces and constraints between tracks in a Markov Random Field and perform joint identification, optimizing the identity assignment over all tracks simultaneously.

1.2. Contributions and outline

The contributions in this thesis are the following:

Chapter 2: Robust face and person tracking Our proposed multi-pose face tracker is robust to diverse conditions and maintains high recall rates. We show in our analysis that our multi-pose tracker has the following advantages: It increases the track recall, *i.e.* more faces/persons are found and can subsequently be identified. Also, we obtain longer tracks on average, which is favorable for identification as the decision can be based on more samples.

Chapter 3: Semi-supervised multi-class learning with constraints For automatic character identification in multimedia data usually all training and test data is available at training time, *i.e.* the full movie or series is available. Therefore, we can model the problem as a semi-supervised/transductive learning problem and leverage unlabeled test data during training. Furthermore, a large number of constraints within and between labeled and unlabeled samples can be automatically acquired. We propose a multi-class learning framework that takes into account all three source of information, (weakly) supervised data, unsupervised data and constraints in a joint formulation.

Chapter 4: Joint multimodal person identification Previous work on naming characters in TV series was limited to naming character appearances with visible (and detected) faces. We extend previous work by identifying person tracks via clothing appearance regardless of the visibility of the face, without the need to manually label neither face nor clothing models. If available, speaker cues from speech recognition and/or subtitles can be integrated as well. Face scores, person scores, speaker presence and constraints are integrated in a global fusion and optimization framework, which jointly optimizes person identity assignments within a shot.

Chapter 5: A Time Pooled Track Kernel Considering the large amount of data (one season alone of a 20-min-per-episode TV series amounts to about 15000 face tracks), efficiency and memory requirements should be taken into consideration. For kernel-based learning methods, memory requirements grow quadratically with the number of features. We therefore use tracks instead of single frames as a basic unit of learning and investigate a family of track-based kernel functions. This results in both a significant reduction in memory requirements and allows to pre-compute a significant part of the required operations for both training *and* testing. A quick train-test turn-around time enables us to quickly re-train and re-test all tracks in an interactive system after small batches of feedback. This allows us to correct the labels of a set of incorrectly classified tracks in about half the time otherwise required without re-training.

Chapter 6 concludes the thesis with a general discussion and outline for future work.

1.3. Previously published contributions

The main contributions of this thesis have been published in different conference proceedings. The basic idea of the tracking approach in Chapter 2 was briefly presented in (Bäuml et al., 2010a). The multi-class classification approach with constraints in Chapter 3 was published in (Bäuml et al., 2013). Chapter 4 has been in part published in (Tapaswi et al., 2012) and has also been in part subject of Makarand Tapaswi's master thesis (Tapaswi, 2011). The track kernel approach (Chapter 5) has been published in (Bäuml et al., 2014).

A full list of my publications, including work not contained in this thesis, can be found in Appendix C.

Chapter 2

Robust Detector-based Tracking

A necessary precursor to visual person identification is localizing the person in the image. A person's position in an image can be defined in different levels of detail, for example by a bounding box around a face, a bounding box around the full person, fine-grained localization of landmarks on the person (*e.g.*, facial landmarks or body joints), or even pixel-wise segmentations of face and body parts.

Depending on the subsequent usage, a simple (and faster to obtain) localization such as a bounding box suffices, for example for marking occurrences of faces in a video or cutting out empty sequences. It can further serve as basis for more fine-grained feature localization. Current state-of-the-art facial landmark localization approaches usually require a rough initial estimate of the location of the face, which a bounding box provides.

In this chapter we address the problem of tracking faces on a bounding box level, *i.e.* the association of such localizations over time. Tracking faces, in contrast to merely detecting them in each frame independently, is beneficial for multiple reasons.

First and foremost, the association of face localizations over time results in *sets* of localizations which can be processed jointly in subsequent steps. For example, identification can be performed using multiple samples instead of just a single one, *e.g.* by fusing the results over the set of samples and thus reducing noise and susceptibility to outliers (*cf*. Fig. 2.1).

Second, tracking can help to filter out false positive localizations. For example, single localizations without support from neighboring frames are likely to be false positives.

Third, by filtering localizations over time, measurement noise can be reduced resulting in more stable and accurate localizations.

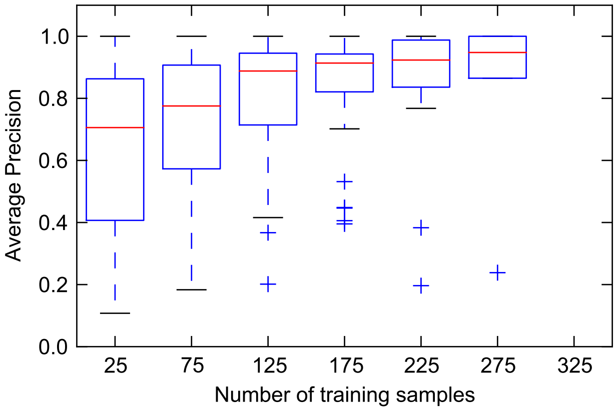


Figure 2.1.: In (Bäuml et al., 2010a) we showed that average retrieval accuracy is higher for longer query tracks, because the number of samples available for training the query model is higher. Obtaining long tracks by a robust tracker is therefore a key factor for high recognition accuracy in videos.

The main objective of our tracker is to increase the recall of tracked faces while keeping a high precision. Since we want to identify faces in subsequent steps, a high recall is paramount. A face that is not localized by the tracker will not be available for identification. Specifically, we address the problem of tracking non-frontal faces with a real-time capable approach. As a byproduct, we perform a rough estimation of the head pose, *i.e.* the angular configuration of the head and face.

2.1. Background and related work

The literature on tracking is vast. In the context of face and person tracking, detector-based tracking is the prevalent approach due to its robustness on real-world data. In the discussion of related work, we will therefore focus on the detector-based tracking of faces and persons.

We can categorize approaches according to the employed appearance model and way of association and filtering.

Appearance model The appearance model encloses the visual knowledge about the tracked object. It generates predictions about the object location, often independently for each frame. Principally, there are two extremes of how much knowledge about the object is known in advance. On the one side, one can have a prior model of appearance, such as a skin-color model or a trained discriminative detector for the object class. In the other extreme, the appearance can be learned during tracking from as little as a single (often manual) initialization. Naturally, many different hybrid models of the above are possible.

Early real-time face and person trackers relied on background subtraction (Wren et al., 1997) and/or color models (Hunke and Waibel, 1994; Yang and Waibel, 1996) to localize the object initially. Both can be seen as using a prior model of appearance (*i.e.*, everything that is not background where the background appearance is learned before; or a previously learned model of skin color). However, by relying on such simple cues, these approaches are very susceptible to changing environmental conditions (*e.g.*, illumination changes, new/moved/removed background objects, similarities in color between foreground and background objects). Also, background subtraction usually relies on a static camera.

Detector-based appearance models With the advent of discriminative object detectors (and increasingly available processing power), object tracking in complex environments shifted towards using such detectors as basis for localizing the object in the image (Andriluka et al., 2008; Babenko et al., 2009; Breitenstein et al., 2009; Dalal and Triggs, 2005; Fröba and Ernst, 2004; Kalal et al., 2010a; Küblbeck and Ernst, 2006; Wu and Nevatia, 2007).

Detector-based tracking builds on two decades of object detection research. Among many ways to localize an object in an image, the now prevalent (and arguably most successful) way is to use a classifier to discriminate between object and non-object instances. The detectors are trained beforehand on a set of positive samples, which contain the object, and negative samples, which do not contain the object. Such classifier-based detectors are much more robust in locating target objects than above-mentioned cues such as background subtraction or color models.

Classification-based object detection approaches can be distinguished by the employed feature(s) and the learning algorithm. For articulated objects, *e.g.* persons, it has proven beneficial to detect parts of the object first, and then merge these part detections for a full object detection (*e.g.*, Andriluka et al. (2008); Bourdev and Malik (2009); Felzenszwalb et al. (2009); Leibe et al. (2008), so here the part selection and merging strategies can also be taken into account.

A key requirement of features to be employed for detection is to be robust against common image variances, *e.g.* to illumination changes, small local deformations or partly occlusions. For face detection, among others Haar-like features (Viola and Jones, 2004) and the Modified Census Transform (MCT) (Fröba and Ernst, 2004; Küblbeck and Ernst, 2006) have been employed successfully. For person detection, shape-based features are more popular, for example edgelets (Wu and Nevatia, 2007) or Histograms of Oriented Gradients (Dalal and Triggs, 2005), which have been shown to work well for other object classes, too (Felzenszwalb et al., 2009). Also, combinations of features are conceivable, for example by selecting the most discriminative features from a feature pool during learning (Dollár et al., 2009; Schwartz et al., 2009).

As learning algorithm, basically any classifier which can distinguish two classes (object vs. background) is conceivable: AdaBoost (Freund and Schapire, 1997) (*e.g.*, for face detection (Küblbeck and Ernst, 2006; Viola and Jones, 2004)), Support Vector Machines (*e.g.*, for generic object detection (Felzenszwalb et al., 2009) or person detection (Bourdev and Malik, 2009; Dalal and Triggs, 2005)), partial least squared regression (*e.g.*, for pedestrian detection (Schwartz et al., 2009)), multiple instance boosting or random ferns (*e.g.*, for model free tracking (Babenko et al., 2009; Kalal et al., 2010a) are just a few examples.

For merging parts, different approaches have been successfully employed. A generalized Hough transform is used by Leibe et al. (2008) to estimate an object center from individual parts, which can be extended with instance-specific models (Seemann et al., 2007). Bourdev and Malik (2009) employ a max-margin variant of the Hough transform (Maji and Malik, 2009) in their Poselet approach. When the part structure is a tree, pictorial structures present an efficient way to compute the MAP configuration of parts (Andriluka et al., 2008; Felzenszwalb et al., 2009). Occlusions can specifically be addressed during part merging by formulating the detection problem as a joint problem for the full image (Wu and Nevatia, 2009).

Track appearance In between a-priori detector-based appearance models and model-free approaches, hybrid approaches have been proposed. The idea is to use a strong prior model to initialize the tracker and/or generate low-level tracklets, and then use higher level appearance models such as color models or instance-specialized detectors (*e.g.*, Ramanan et al. (2007b)) to continue tracking the objects or linking low-level tracklets.

Kuo and Nevatia (2011) use tracklet-specific appearance models for linking different tracklets, which Yang and Nevatia (2012b) extended to part-based appearance models. Since each tracklet stems from detections, the appearance model does not cause drift. On the other hand, apart from interpolated frames between linked tracklets, this approach will not find objects that are not detected by the detector. Also, it can lead to track switches, if for example persons wear similar clothing in the scene.

In order to improve over interpolation between tracklets, Sharma et al. (2012) propose to adapt the underlying detector by unsupervised collection of training data. Missing detection windows obtained by interpolation and low confidence detections which overlap with interpolated tracklets are used as positive training data, non-tracked detections are negative training data. Similar to (Babenko et al., 2009), the detector adaptation is formulated as a multiple instance learning problem to mitigate for possible spatial errors.

Association The primary goal of association is to assign measurements (*e.g.*, detections) to an existing track hypothesis of the same target. Association is usually based on a combination of an appearance model (see above), a motion model (*e.g.*, Huang et al. (2008); Isard and Blake (1998b)) and possibly additional cues such as homography information between cameras and/or the ground plane (*e.g.*, Hofmann et al. (2013); Khan and Shah (2008)).

Many tracking approaches assume only knowledge about the current and one or more previous frames (*e.g.*, Breitenstein et al. (2009)). Such an assumption is necessary for real-time and online tracking approaches, where a state estimation is expected for each frame without waiting for more frames into the future. However, if for example ambiguities in appearance cannot be resolved based on the information given in the current frame, this can lead to identity switches. In part, this can be resolved by looking at more than just the current frame (Leibe et al., 2007). Assuming prior knowledge of the *full* video, tracking can be reduced to a global association problem, determining which detections form together a track of one person. Different strategies for solving such global association problems exist, for example shortest-paths searches (Berclaz and Fleuret, 2011), linear programming (Jiang et al., 2007), min-cost flow (Hofmann et al., 2013; Zhang et al., 2008), greedy forward-backward tracking (Wu and Nevatia, 2007) or hierarchical association (Huang et al., 2008).

In association-based tracking (*e.g.*, Huang et al. (2008); Li et al. (2009); Wu and Nevatia (2007, 2006); Yang and Nevatia (2012b)) given detections are associated to tracklets based on an affinity measure. The affinity measure can for example comprise manually defined distances between location, size and appearance between two detections or tracklets (Huang et al., 2008; Wu and Nevatia, 2007). It can also be learned from training data (Li et al., 2009) or can be based on the number of consistent KLT-tracked features between two detections (Everingham et al., 2006; Sivic et al., 2009).

Association-based tracking is especially appealing when the full video data is available beforehand (Li et al., 2009; Wu and Nevatia, 2007; Yang and Nevatia, 2012b), since at each step the best pair of detections/tracklets can be associated without being restrained to the current time step. Remaining gaps between tracks can be bridged for example using mean shift (Wu and Nevatia, 2007), or by a hierarchy of association steps using increasingly complex affinity models (Li et al., 2009; Yang and Nevatia, 2012a).

Graph-based approaches model the tracking problem as a graph with detections as nodes and weighted edges between them (Berclaz and Fleuret, 2011; Hofmann et al., 2013; Salvi et al., 2012). In contrast to association-based approaches, which iteratively make greedy decisions, a global solutions can be found by shortest-paths searches (Berclaz and Fleuret, 2011), linear programming (Jiang et al., 2007) or min-cost flow (Hofmann et al., 2013; Zhang et al., 2008), usually at the expense of higher computational cost.

Online and real-time tracking In many real world scenarios, a tracker is required to operate in real-time and in an online fashion, *i.e.* it cannot require information about future frames. In order to achieve real-time performance, a tracking step should last well below 1s. For example, for a video with 25fps, each processing step should require less than 40ms.

When computational power was scarce, this required either simple models (Yang and Waibel, 1996) or hardware specific implementation tuning (Gavrila and Philomin, 1999). However, due to progress in hardware capabilities and object detection, there exist several discriminative detector approaches today that are real-time capable on commodity hardware (*e.g.*, Küblbeck and Ernst (2006); Viola and Jones (2004)).

Another technique to speed up the underlying detection step is to run the detector only at selected regions. For example, one can perform foreground-background segmentation and run detector(s) only on foreground regions (Nechyba et al., 2008). Similarly, difference images can be used instead of foreground segmentations.

Other speedups target the feature computation step of the detector, and approximate feature responses at multiple scales from a few scales (Dollár et al., 2010). Benenson et al. (2012) enhance this further by learning multiple, specialized detector models for different scales.

Joint face tracking and head-pose estimation By determining the pose of the head and face, one can estimate the viewing direction of a person and thus get a first estimate of the focus of attention (Stiefelhagen, 2002; Voit and Stiefelhagen, 2006).

Most approaches for head pose estimation require a prior localization of the face or head. For simplicity, face tracking and head pose estimation are often performed in sequence (*e.g.*, Voit and Stiefelhagen (2005)).

Considering both problems jointly can lead to an increase in both tracking and head pose estimation accuracy. Pose estimation or verification can influence the tracking by including pose as a continuous (*e.g.*, Krahnstoever et al. (2011)) or discrete (*e.g.*, Ba and Odobez (2004)) variable in a particle state. (Kim et al., 2008) only include in-plane rotation in the particle state, but model yaw changes as a constraint by including the minimum distance to any pose subspace in the scoring of the particle.

An important factor is the approach for determining the head pose. Krahnstoever et al. (2011) use a face detector. Neural networks (Voit and Stiefelhagen, 2005) or PCA subspaces (Kim et al., 2008), which only discriminate between poses but not against non-faces, are also possible choices. Furthermore, estimates from multiple cameras can be combined (Voit and Stiefelhagen, 2005, 2006)

Ba and Odobez (2004) integrate face tracking and exemplar-based head pose estimation in a probabilistic particle filter-based framework. The particle state-space includes – in addition to the position – the face's pan angle θ. Gaussian mixture models for 9 equally-spaced pan angles are learned on a training data set, and particles are scored with the head pose mixture corresponding the current state.

Chen et al. (2011) estimate full body orientation instead of head pose. The detector response of the pose class closest to the particle's orientation state is used to score the particle. Since the body pose restricts possible head orientations, jointly tracking location, head pose and body orientation can further increase performance. Krahnstoever et al. (2011) include both body and head orientation in the particle's state. For both body and face orientation scoring a detector is used. Both above approaches perform tracking in ground plane coordinates.

2.1.1. Discussion and contribution

Since our object class is known a-priori (*i.e.*, faces), we can leverage pre-learned appearance models. Drawing from the work of two decades on object classification and detection, a discriminative classifier, *i.e.* a face detector, can be trained in advance. For robustness at the detection step, invariant features to possible image changes such as illumination are preferred (*e.g.*, gradients, LBP, MCT), whereas color models are insufficient for the initial localization (face color is not well defined when working for example with low light images). Due to the rigid and discriminative nature of faces, face detectors are among the best performing object detectors in the literature. Nevertheless, pose changes induce substantial appearance changes and should be addressed either by part-based detectors, or detector-banks for a set of pose classes. The integration of multiple detectors as appearance model into the tracker without the computational burden of running all detectors at the every frame is key to a real-time capable tracker.

A particle filter has been shown to work well with a detector-based appearance model. While keeping independent particle filters for each track is paramount to prevent an exponential explosion of the state space, care has to be taken in handling occlusions to avoid track switches.

In the literature the focus for joint face and head pose tracking usually lies on improving the precision of the head pose estimation. While the estimation of the head pose is a byproduct of our approach, we are more interested in improving the recall of tracked faces by tracking over different poses. Nevertheless, it can be useful for potential higher applications that utilize or depend on head pose. An approach close to ours is presented by Ba and Odobez (2004). However, there a generative model Gaussian mixture model based on 4 different filter responses is used to determine the likelihood the discretized head pose, which is likely to fail on realistic multimedia data.

Some of the discussed approaches use information about the ground plane (*e.g.*, Chen et al. (2011); Krahnstoever et al. (2011); Morzinger et al. (2011)), which makes these approaches infeasible where information about the ground plane is not available. In this work, we only assume to be working with a single camera, which is the case for the application scenario of multimedia data. Nevertheless, a monocular tracker can always be applied in multi-camera settings, too, such as a camera network.

Contributions We extend the work on detector-based tracking with a particle filter with the integration of multiple independent face detectors for different pose classes without the need to run all detectors for each particle. We annotated a large data set based on two TV series comprising more than annotated 100000 faces, allowing to evaluate the tracker on a diverse set of conditions. On this data set, we evaluate our approach with increasing number of detectors and demonstrate that our proposed approach reduces the number of misses by up to 50%, leading to an consistent increase of Multiple Object Tracking Accuracy (MOTA) of 15% points compared to a frontal-face-only tracker.

2.2. Robust online multi-pose detector-based tracking

We formulate object tracking as a sequential state estimation problem. Our goal is to estimate the object state $\mathbf{x}_t \in \mathbb{R}^n$ at time t, where the state for example consists of position and size of the object in the image. We assume that the object state progresses according to the system model

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{w}_t) \tag{2.1}$$

and can be observed via the observation model

$$\mathbf{z}_t = g(\mathbf{x}_t, \mathbf{v}_t) \tag{2.2}$$

where f and g are in general non-linear functions $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\mathbf{w}_t, \mathbf{v}_t \in \mathbb{R}^n$ denote state-independent process and observation noise, respectively, from a zero-mean normal distribution. Thus, the state follows a first order Markov process, *i.e.* f only depends on the state of the previous time step $t-1$.

For detector-based tracking, evaluating g usually involves evaluating the object detector at hundreds of thousands of windows of an image pyramid. However, this is computationally expensive, especially when employing multiple object detectors to account for object variability (the evaluation time of n independent detectors is usually linear in the number of detectors). We address this problem by reducing the number of image windows where object detectors are evaluated to estimate the observational density $p(\mathbf{z}_t|\mathbf{x}_t)$ by means of a particle filter.

2.2.1. The bootstrap filter

The *bootstrap filter* (Gordon et al., 1993), also known as the *CONDENSATION* algorithm (Isard and Blake, 1998a), recursively estimates the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_0,\ldots,\mathbf{z}_t)$ by iterative prediction and update steps. The density of the state $\mathbf{x}_t$ is approximated by a set of N *particles*, *i.e.* weighted samples $Q_t = \{(\mathbf{q}_t^{(k)}, w_t^{(k)}), k = 1,\ldots,N\}$. In the prediction step, the particles are first resampled and then progressed according to the system model. The update step then reweights the particles according to the observational density $p(\mathbf{z}_t|\mathbf{x}_t)$.

The resampling assures that particles are evenly distributed according to the density $p(\mathbf{x}_t)$. This is achieved by sampling with replacement N new particles Q_t from Q_{t-1}. The probability to sample particle $\mathbf{q}_{t-1}^{(k)}$ is proportional to its weight $w_{t-1}^{(k)}$, *i.e.* $P_Q(k=i) = \frac{w_{t-1}^{(i)}}{\sum_j w_{t-1}^{(j)}}$.

The prediction updates the particles to approximate the prior distribution at the next time step t: $p(x_t|\mathbf{z}_0,\ldots,\mathbf{z}_t) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) \cdot p(\mathbf{x}_{t-1}|\mathbf{z}_0,\ldots,\mathbf{z}_t) d\mathbf{x}_{t-1}$. This is achieved by updating the state of each particle according to the system model with

$$q_t^{(k)} = f(q_{t-1}^{(k)}, w_t)\,. \tag{2.3}$$

Finally, in the update step the particle weights are set according to the observations z_t. In order to maintain $\sum_k w_t^{(k)} = 1$, the weight update is normalized:

$$w_t^{(k)} = \frac{p(z_t|q_t^{(k)})}{\sum_j p(z_t|q_t^{(j)})} \tag{2.4}$$

Since the number of particles does not change in the resampling step, the computational requirements for updating the weights do not change over time. By choosing the number of particles N accordingly, we can trade off accuracy in the approximation of the density versus computational requirements for obtaining the observations.

In contrast to the Kalman filter (Kalman, 1960), neither f nor g have to be linear, and the propagation of the covariance is not explicitly required. More importantly, for a Kalman filter it is not as straightforward to control the computational requirements at the observation step, which constitutes the main computational burden in detector-based tracking.

2.3. Detector-based face tracking with the bootstrap filter

In order to perform detector-based tracking using the bootstrap filter, the elements of the object state, the system model and the observation model must be specified, which we will do in the following. Although the bootstrap filter is capable of modeling multi-modal distributions, we choose to model *each* track with its own bootstrap filter and consider interactions between tracks only during the scoring of particles (Lanz, 2006). Thus, the track state is defined for a single track only. Interactions between tracks are handled by the occlusion model.

State We include the position (x, y) in pixels, scale s and orientation (ψ, θ, ϕ) of the face in the state:

$$q_t = (x, y, s, \psi, \theta, \phi) \tag{2.5}$$

The scale s is inversely proportional to the size of the face in the image, relative to the detector base size. We do not want to require any camera calibration, therefore we cannot determine the exact relative 3d position of the face to the camera. However, s is proportional to z, the true distance of the face to the camera. To a certain degree, we therefore can still reason about distances and sizes of objects relative to the camera. ψ, θ and ϕ correspond to the *yaw*, *pitch* and *roll* angles of the head pose, respectively. By including the head pose, we are able to select a single specific face detector to score the current state.

We determine the current position of a track as weighted average over all particles

$$\begin{aligned} \mathbf{z}_t^* &= \sum_k w_t^{(k)} q_t^{(k)} && (2.6) \\ &= \sum_k w_t^{(k)} (x, y, s, \psi, \theta, \phi)_t^{(k)} \quad , && (2.7) \end{aligned}$$

assuming that our particle distribution is unimodal. The unimodality is softly enforced by tracking each face by its own set of particles and explicitly handling occlusions (see below, *Occlusion handling*). Alternatively, the current position could be estimated as the mode of the state distribution, *e.g.* via mean shift. However, we found that in practice the simple weighted average is sufficient.

The weighted average of the particle status gives us a rough localization of the face and estimation of the head pose. We also compute a rough position estimate of facial landmark positions. To this end, we first estimate for each detector its mean facial landmark positions relative to its detection window on a separate training set. We then compute the weighted average of the facial landmark positions over all particles. A particle's mean landmark position is determined from its corresponding detector, the one with the closest pose class to the particle's state (*cf*. below, *Observation Model*). This initial estimate is surprisingly accurate and subsequently used to initialize a dedicated landmark detection method (see Sec. 3.3.2).

Prediction The system model describes the expected motion of the face given the state of the last frame. We use a stationary system model. Motion of the face is modeled within the process noise $\mathbf{w}_t$, therefore the prediction step diffuses the last state according to $\mathbf{w}_t$ with

$$(x,y)_t = (x,y)_{t-1} + \frac{1}{s_t}(w_x, w_y)_t \tag{2.8}$$

$$s_t = s_{t-1} + w_{st} \quad . \tag{2.9}$$

$\mathbf{w}_t$ is drawn from a zero mean Normal distribution with different variances for each of the dimensions. The influence of the noise for (x,y) needs to be scaled by the size of the face, since a larger face can move greater distances in terms of pixels in the image. Since the scale s is inversely proportional to the size of the face, it is included as $\frac{1}{s}$ in the noise term for (x,y).

Similarly, the system model for the head pose is also stationary:

$$(\psi,\theta,\phi)_t = (\psi,\theta,\phi)_{t-1} + (w_\psi, w_\theta, w_\phi)_t \tag{2.10}$$

Here, the variance is independent of the scale. In theory, the angular noise terms should follow a *von Mises-Fisher* distribution (Mardia and Jupp, 1999). However, since we are tracking faces, not heads, we neglect the wrap-around of the angles here and approximate the distribution by a Normal distribution.

Observation model For updating the particles' weights, *i.e.* estimating $p(z_t|q_t)$, we employ a bank of face detectors. Each face detector is a boosted cascade of weak classifiers based on MCT features (Küblbeck and Ernst, 2006). We trained detectors at multiple yaw and roll angles to account for different head poses. Each detector provides a calibrated confidence value in the range $[0,1]$. Only valid detections which passed the full detector cascade of N stages have a confidence > 0.

For each particle $(\mathbf{q}_t^{(k)}, w_t^{(k)})$, we determine the detector with the pose class γ closest to the particles pose state $(\psi, \theta, \phi)_t^{(k)}$. The particle's weight is then updated according to

$$w_t^{(k)} = \begin{cases} 0 & \text{if} \quad n < N \\ \sigma\left(\theta_1^{(\gamma)} H_N^{(\gamma)}\left(\mathbf{f}_t^{(k)}\right) + \theta_0^{(\gamma)}\right) & \text{if} \quad n = N \end{cases} \tag{2.11}$$

where $\sigma(x) = (1+e^{-x})^{-1}$ is the sigmoid function and n the number of passed stages (from a total number of N stages) in the detector. $H_N^{(\gamma)}\left(\mathbf{f}_t^{(k)}\right)$ denotes the stage sum of the last cascade stage of the detector for pose class γ evaluated on the feature vector f extracted at location $\mathbf{x}_t^{(k)}$. θ_0 and θ_1 are calibration parameters estimated on a validation set for calibrating the face detector scores to a common range of $[0,1]$. While using stage-sums directly is also possible, the value range of stage-sums is not necessarily comparable across different detectors. By calibrating the detector scores to $[0,1]$ we reduce a systematic bias towards any of the pose classes.

Occlusion handling The above observation model does not uniquely associate a detection with one track only. If the position and extent of particles of two tracks overlap, they can score detections from the same underlying face, for example when one face is occluded by another face. This double counting is undesired for two reasons. First, in the event of no other detection, the occluded track will be drawn to the occluding face, overlapping the "true" track for this face. Second, the detection would reset $n_{term} = 0$ for the occluded track, avoiding that the occluded track terminates.

Both problems are solved by the following heuristic. The weights of all particles that are closer than θ_{occ} to the mean of another track, are set to $\epsilon = e^{-12}$.

Similarly, the image border can be seen as an occlusion. We therefore also set the weights of all particles outside the image borders to $\epsilon = e^{-12}$.

Initialization In order to start a new face track, we scan the full frame with a subset of the detector bank every k frames. The value of k trades off the average time until a new face is detected versus the computational load induced by the scan. A good value for k is usually application dependent, but $k = 5$ for a 25fps sequence is a good default for an interactive application (corresponding to one scan every 0.2 seconds). In order to avoid the latency induced by the computational expensive scan every k frames, we smooth out the scan over several frames by processing only parts of the full pyramid in each frame similar to Küblbeck and Ernst (2006).

Validation A new track is initialized from each detection. Although the employed MCT face detectors exhibit a low false positive rate, false positive detections cannot be disregarded. Often, they only occur in one frame, but not consecutive frames. Also, a true face results in usually more than one detection due to the shift and scale tolerance of the detector. Therefore, we require that a track accumulates at least n_{init} detections before it becomes a *valid* track, *i.e.* for n_{init} particles the respective detector cascade has to complete all N stages, possibly spread out over multiple frames. Tracks which are terminated (see below) without being validated are discarded. A lower n_{init} reduces the number of missed tracks, while on the other hand leading to a higher number of spurious tracks stemming from false positive detections. A separate validation step such as a confident detection of facial landmarks or size-based reasoning could also help filtering out possible false positive frames (Tapaswi et al., 2014c).

Termination A track should be terminated when the face is no longer present in the image, for which a good indication is that none of the particles score any detections. However, it is common that from time to time none of the particles detect a face due to image noise (*e.g.*, compression artifacts) or the random nature of the particle distribution, whereas in the following frame the face is detected again. We therefore terminate a track only if no detections were made in any of the particles for more than n_{term} frames. A high n_{term} leads to longer, continuous tracks, which can bridge short occlusions or frames in which the detector could not find a face, *e.g.* due to consistent noise or a difficult face pose. However, it is also more likely that the track switches from a face that is no longer present to the face of another person. A low n_{term}, possibly even 0, leads to fewer false positive track frames and shorter but possibly fragmented tracks. For identification, we prefer the latter to avoid track switches, since we usually infer the identity from the whole track, assuming that all frames of the track belong to the same person.

2.4. Evaluation

We evaluate the tracker on a large multimedia data set comprising 12 episodes of two different TV series. We first briefly describe the employed evaluation metrics (Sec. 2.4.1), then introduce our collected data set (Sec. 2.4.2) and pre-processing steps (Sec. 2.4.3) . Finally, we analyze different aspects of the tracker and compare against other approaches from the literature (Sec. 2.4.4).

2.4.1. Evaluation metrics

We evaluate tracking performance using the CLEAR Multiple Object Tracking (MOT) metrics (Bernardin and Stiefelhagen, 2008). They are designed for evaluating *identity* tracking of multiple objects, *i.e.* they capture not only whether the location of an object is correctly traced over time, but also whether its identity is maintained. This is of particular importance for face and person tracking, since we will assume later for *identification* that all frames within a track belong to the same person.

The Multiple Object Tracking Precision is defined as

$$\mathrm{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} , \tag{2.12}$$

where d_t^i is the distance of hypothesis i to the matched ground truth object, and c_t the number of matches found for time t.

The Multiple Object Tracking Accuracy is defined as

$$\mathrm{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mm_t)}{\sum_t g_t} , \tag{2.13}$$

where m_t is the number of misses at time t, fp_t the number of false positives (*i.e.* unmatched hypotheses), and mm_t the number of mismatches (*i.e.* correspondences that do not maintain a previous object identity mapping).

MOTA	Multiple Object Tracking Accuracy	$1 - \frac{\sum_t (m_t + fp_t + mm_t)}{\sum_t g_t}$
MOTP	Multiple Object Tracking Precision	$\frac{\sum_{i,t} d_t^i}{\sum_t c_t}$
FP	Number of False Positives	$\sum_t fp_t$
FPR	False Positive Rate	$\frac{\sum_t fp_t}{\sum_t g_t}$
MISS	Number of Misses	$\sum_t m_t$
MISSR	Miss Rate	$\frac{\sum_t m_t}{\sum_t g_t}$
MM	Number of Mismatches	$\sum_t mm_t$
MMR	Mismatch Rate	$\frac{\sum_t mm_t}{\sum_t g_t}$
GMM	Number of Good Mismatches	$\sum_t gmm_t$
GMMR	Good Mismatch Rate	$\frac{\sum_t gmm_t}{\sum_t g_t}$
BMM	Number of Bad Mismatches	$\sum_t bmm_t$
BMMR	Bad Mismatch Rate	$\frac{\sum_t bmm_t}{\sum_t g_t}$
TR	Track Recall	$\frac{t_1}{T}$
TP	Track Precision	$\frac{t_1}{h_0 + t_1}$

Table 2.1.: Employed abbreviations in the result tables in this and the following chapters.

It is important to note that the average is computed over all frames, not individually for each frame, and only then averaged over all frames in a second step.

For our purposes, MOTA is the more important measure compared to MOTP, since a precise localization of the bounding box is not important for identification. We estimate more detailed positions of facial landmarks in a separate step anyway for a fine-grained alignment. The estimated landmarks do not necessarily coincide with the labeled ground truth bounding boxes.

In order to be able to analyze sources of errors, we also further split up MOTA and report miss rate, false positive rate and mismatch rates individually. Mismatches can be further divided into *good* (gmm_t) and *bad* (bmm_t) mismatches. A good mismatch occurs when the tracker ends a track, *e.g.*, due to a partial occlusion, and then continues to track the person afterwards, albeit under a different temporary identity. A bad mismatch on the other hand occurs, when a track switches from one underlying person to another. Such errors are *bad* because they violate our assumption that a track follows only one unique person and we can assign one identity to it during identification.

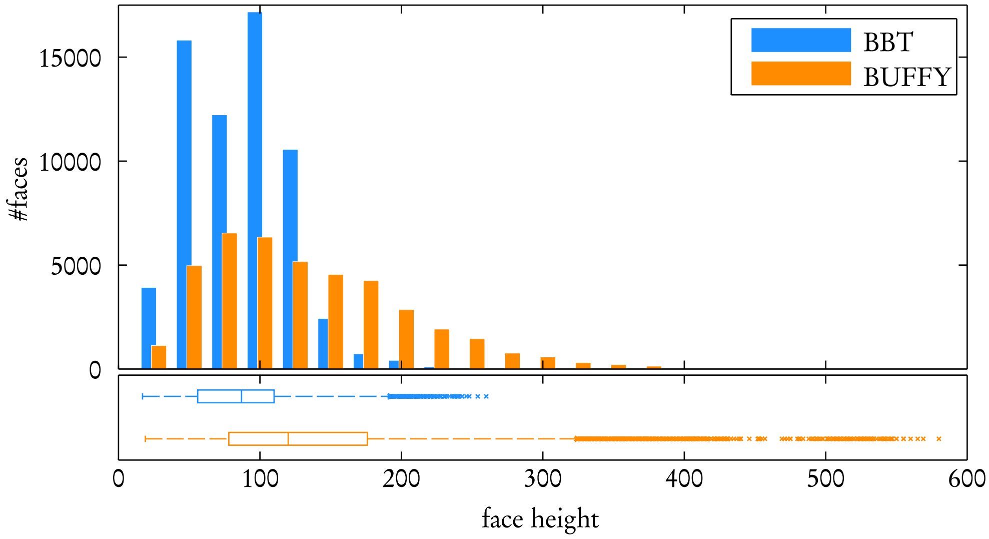


Figure 2.2.: Comparison of the two series' face size distributions. BUFFY contains more large faces due to many close-up face shots. The face size distribution has a long tail with faces up to 580px in height.

We further report track recall and track precision, which give an intuition on the number of tracks which are missed completely, and the track-level false-positive rate, respectively. Let T be the total number of tracks, t_1 the number of tracks for which at least one hypothesis matches (true positives), and h_0 the number of hypotheses that do not match a track (false positives). Then track recall is defined as $TR = \frac{t_1}{T}$ and track precision as $TP = \frac{t_1}{h_0+t_1}$.

In the result tables in Sec. 2.4.4 we will use the abbreviations from Tbl. 2.1.

Ep.	#tracks	#faces	#DCOs	avg. height [px]	avg. length [frames]
BBT Season 01					
1	641	11284	1711	84.8	81.3
2	630	9334	702	92.8	67.6
3	702	10724	1335	84.6	71.4
4	656	9892	1533	82.4	70.4
5	617	9380	1303	91.5	71.0
6	852	12761	1547	82.2	69.9
BUFFY Season 05					
1	785	6141	220	140.1	68.2
2	943	7343	170	128.1	67.9
3	1158	7415	256	121.8	54.0
4	864	6678	343	143.6	67.3
5	854	6318	288	149.3	64.0
6	1102	7609	244	130.5	59.0
Sum	9804	104879	9652	–	–

Table 2.2.: Ground truth statistics for face tracking evaluation on BBT and BUFFY data set. The average face size in BBT is lower than in BUFFY due to different filming styles. Note that, while the number of tracks in BUFFY is slightly higher than in BBT, the overall number of annotated faces is lower due to only annotating every 10th frame compared to every 5th for BBT.

2.4.2. Data set

We evaluate our tracking approach on a data set consisting of 6 episodes each of the two TV series *The Big Bang Theory* (BBT, episodes S01E01-06) and *Buffy the Vampire Slayer* (BUFFY, episodes S05E01-06).

We annotated the data set with face bounding boxes and identities. Due to the different episode lengths (BBT: ≈20min, BUFFY: ≈45min) we annotated every 5th frame for BBT and every 10th frame for BUFFY to achieve full coverage on both series with approximately the same labeling effort. Evaluation is performed only on annotated frames, *i.e.* we *do not* perform interpolation of ground truth to avoid possible errors. Faces which are poorly illuminated, strongly blurred, largely occluded or with more than profile head pose, were marked as *don't care objects* (DCOs). DCOs will not be counted as errors if missed, but similarly not as error if actually localized by the tracker.

The minimum annotated face is about 18px in width. Due to different filming styles, the distribution of face sizes is different for the two series (see Figure 2.2). BUFFY's face size distribution is skewed towards larger faces, owing to many close-up face shots. The maximum face size is $580px$ versus $260px$ for BBT.

A brief overview over the statistics of the data set is given in Tbl. 2.2. A total of 104879 faces and 9652 DCOs have been annotated. On average, there are 683 tracks per episode in BBT, and 951 tracks per episode in BUFFY, totaling up to 9804 tracks for all 12 episodes. To the best of our knowledge, this makes this data set one of the largest available data sets for the evaluation of face tracking and the largest in the context of multimedia data.

2.4.3. Preprocessing

Multimedia data usually consists of a series of *shots*, *i.e.* a sequence of frames that are filmed uninterrupted from one camera. At a shot boundary, tracks end due to the switch of camera/viewing angle. We therefore detect such shot boundaries explicitly before tracking and forcefully terminate tracks at each of them.

Shot boundary detection We detect shot boundaries using the *Displaced Frame Differences* (Yusoff et al., 1998) of motion compensated consecutive frames. We divide the image into equally sized blocks (*e.g.*, $16x16px$), and compute the Displaced Frame Difference at timestep t as

$$DFD(t) = \sum_{x,y} ||I_{16x16}((x,y),t) - I_{16x16}((x,y) - o(x,y,t), t-1)|| \quad , \tag{2.14}$$

where $I_{16x16}((x,y),t)$ is the image block of size $16x16px$ at position (x,y) and timestep t. $o(x,y,t)$ denotes the motion offset of the image block from the previous to the current frame, *e.g.* determined by block matching. We filter the obtained DFD feature by a series of open and close operations to remove noise and expose the true peaks, which are then thresholded to determine the shot boundary locations.

With this method, we correctly detect 1981 shot boundaries on the first 6 episodes of BBT, with 2 misses and 8 false positive detections. Misses can be considered more severe than false positives, since they might result in a track switch if – by chance – faces of different persons are nearby in old and new shot. A continuation of the track from one person to another violates our assumption on a unique identity within a track. On the other hand, a false positive “only” splits a track in two parts, which will have to be identified later independently. However, with a total of 10 failure cases compared to 1981 correct detections, the induced errors are negligible.

2.4.4. Results and analysis

Our motivating thesis was that by employing multiple face detectors for different poses, we achieve better tracking performance, especially a higher track recall. We therefore start by investigating the influence of including non-frontal detectors in the tracking procedure.

Frontal vs. full pose We analyze the dependency of tracking performance using an increasing amount of non-frontal detectors. A frontal-only tracker serves as baseline. This frontal-only baseline is similar to the approach described in (Küblbeck and Ernst, 2006), except that we use a particle filter instead of a Kalman filter.

See Fig. 2.3 for different performance measures in dependency of the maximum yaw angle of the employed detectors. MOTA increases consistently over different episodes from the frontal-only case to using detectors of up to 90 degree yaw angle. As expected, this is mainly dominated by a reduction in miss rate, *i.e.* more true faces can be localized

Ep.	max. yaw	MOTA	MOTP	FP(R)	MISS(R)	BMM(R)	GMM(R)	TR	TP
1	0	0.665	0.732	599 (0.058)	2744 (0.267)	4 (0.000)	88 (0.009)	0.737	0.824
	45	0.731	0.719	356 (0.035)	2300 (0.224)	1 (0.000)	102 (0.010)	0.765	0.898
	90	0.817	0.716	655 (0.064)	1128 (0.110)	1 (0.000)	96 (0.009)	0.854	0.864
2	0	0.480	0.640	1081 (0.125)	3327 (0.384)	1 (0.000)	96 (0.011)	0.677	0.694
	45	0.583	0.630	610 (0.070)	2889 (0.333)	2 (0.000)	114 (0.013)	0.728	0.824
	90	0.674	0.615	790 (0.091)	1940 (0.224)	4 (0.000)	96 (0.011)	0.820	0.801
3	0	0.396	0.679	2423 (0.230)	3856 (0.366)	5 (0.000)	91 (0.009)	0.652	0.622
	45	0.513	0.664	1315 (0.125)	3715 (0.352)	2 (0.000)	100 (0.009)	0.658	0.756
	90	0.617	0.638	1736 (0.165)	2211 (0.210)	0 (0.000)	97 (0.009)	0.782	0.745
4	0	0.595	0.697	963 (0.100)	2846 (0.296)	2 (0.000)	86 (0.009)	0.684	0.758
	45	0.687	0.684	473 (0.049)	2457 (0.255)	2 (0.000)	78 (0.008)	0.702	0.849
	90	0.767	0.663	784 (0.082)	1378 (0.143)	2 (0.000)	74 (0.008)	0.800	0.806
5	0	0.562	0.673	1031 (0.113)	2888 (0.316)	2 (0.000)	76 (0.008)	0.623	0.715
	45	0.638	0.656	747 (0.082)	2493 (0.273)	1 (0.000)	61 (0.007)	0.649	0.795
	90	0.712	0.635	1203 (0.132)	1355 (0.148)	7 (0.001)	64 (0.007)	0.793	0.772
6	0	0.489	0.666	1808 (0.142)	4542 (0.358)	3 (0.000)	127 (0.010)	0.672	0.705
	45	0.578	0.649	1072 (0.084)	4152 (0.327)	3 (0.000)	125 (0.010)	0.672	0.791
	90	0.682	0.635	1618 (0.128)	2306 (0.182)	3 (0.000)	112 (0.009)	0.803	0.740

Table 2.3.: Detailed tracking results for BBT S01E01-06 in dependency of the maximum yaw angle of the employed detectors. A consistent increase in MOTA by over 0.15 can be observed in all 6 episodes. The number of misses is reduced by about 50%, while at the same time not allowing significantly more false positives. Consequently, the track recall increases consistently over all episodes, reaching about 80% across all episodes.

which cannot be found with a frontal detector only. The false positive rate increases, but at a much slower rate than the reduction in miss rate, leading to an overall increase in MOTA.

For detailed comparisons between the baseline and a full-pose tracker on BBT and BUFFY see Tables 2.3 and 2.4, respectively. MOTA increases consistently by over 0.15. Here again, we see the strong reduction in miss rate as in Fig. 2.3. In contrast, the false positive rate increases only moderately. In coherence with the reduced miss rate, track recall (TR) increases, while track precision remains stable. This confirms that by employing non-frontal detectors, we are able to find more of the existing faces which was our primary goal.

In Tbl. 2.5 we analyze the average track length in dependency of the maximum yaw angle of the detectors. The average track length increases consistently when adding

Ep.	max. yaw	MOTA	MOTP	FP(R)	MISS(R)	BMM(R)	GMM(R)	TR	TP
1	0	0.615	0.751	662 (0.111)	1532 (0.256)	3 (0.001)	106 (0.018)	0.858	0.780
	45	0.687	0.732	332 (0.056)	1435 (0.240)	1 (0.000)	104 (0.017)	0.863	0.891
	90	0.750	0.714	392 (0.066)	1014 (0.170)	1 (0.000)	91 (0.015)	0.896	0.854
2	0	0.521	0.733	1115 (0.155)	2102 (0.292)	8 (0.001)	233 (0.032)	0.823	0.736
	45	0.606	0.739	702 (0.097)	1893 (0.263)	7 (0.001)	237 (0.033)	0.835	0.846
	90	0.707	0.730	841 (0.117)	1058 (0.147)	6 (0.001)	211 (0.029)	0.885	0.809
3	0	0.479	0.734	1107 (0.153)	2444 (0.339)	6 (0.001)	206 (0.029)	0.780	0.739
	45	0.568	0.727	614 (0.085)	2288 (0.317)	6 (0.001)	212 (0.029)	0.784	0.846
	90	0.656	0.698	874 (0.121)	1403 (0.194)	7 (0.001)	203 (0.028)	0.876	0.809
4	0	0.568	0.725	570 (0.087)	2091 (0.320)	2 (0.000)	163 (0.025)	0.804	0.794
	45	0.629	0.713	365 (0.056)	1890 (0.289)	1 (0.000)	170 (0.026)	0.812	0.861
	90	0.715	0.697	534 (0.082)	1169 (0.179)	2 (0.000)	160 (0.024)	0.880	0.838
5	0	0.588	0.756	802 (0.129)	1637 (0.262)	1 (0.000)	128 (0.021)	0.819	0.761
	45	0.671	0.748	482 (0.077)	1435 (0.230)	1 (0.000)	135 (0.022)	0.835	0.844
	90	0.721	0.725	587 (0.094)	1016 (0.163)	2 (0.000)	135 (0.022)	0.888	0.844
6	0	0.593	0.726	793 (0.105)	2113 (0.281)	3 (0.000)	156 (0.021)	0.843	0.794
	45	0.651	0.730	502 (0.067)	1971 (0.262)	3 (0.000)	155 (0.021)	0.847	0.862
	90	0.706	0.719	773 (0.103)	1275 (0.169)	1 (0.000)	166 (0.022)	0.899	0.824

Table 2.4.: Detailed tracking results for BUFFY S05E01-06 in dependency of the maximum yaw angle of the employed detectors. Similar to results from BBT, MOTA and track recall consistently increase when including increasingly non-frontal detectors. A notable difference to BBT is that the track recall almost reaches 90% despite higher miss rates. While track coverage is better, the increased good mismatch rates suggest that for BUFFY, more tracks are split in two or more independent tracks – possibly because of more difficult image conditions – and only intermediate frames are missed.

increasingly out-of-plane detectors. In addition to the higher track recall, we also obtain *longer* tracks, which is beneficial for subsequent identification as we motivated in the introduction of this chapter.

Further notable as a general property of the tracker is the low number of bad mismatches across all episodes. The reported mismatch rates are largely dominated by good mismatches, which are tolerable for identification as discussed in Sec. 2.4.1.

Track validation and termination In Fig. 2.4 we evaluate the influence of n_{init} and n_{term}. The respective metrics are averaged over the 6 episodes BBT S01E01-06. As expected, a higher n_{init} reduces the false positive rate since a track needs to accumulate more detection evidence before being validated. A lower n_{term} ensures that tracks end

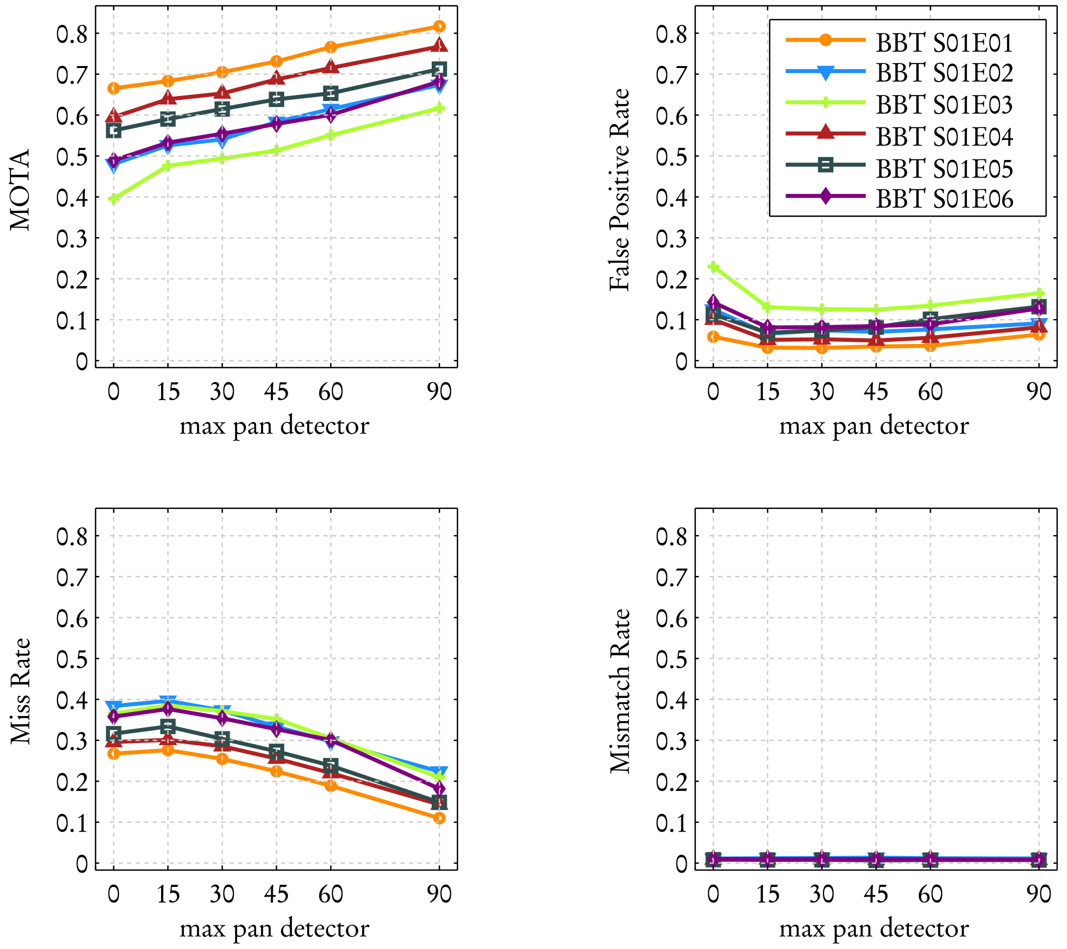


Figure 2.3.: Tracking performance with increasingly out-of-plane rotated detectors on BBT. The x-axis denotes the maximum yaw angle of the underlying detectors, *e.g.*, at $x = 30$ the tracker uses detectors for yaw angles of 0, 15 and 30 degrees (inclusive). The miss rate decreases consistently over all episodes, while the false positive rate increases, although far slower than the miss rate. Consequently, MOTA increases consistently when adding more non-frontal detectors. The corresponding plot for BUFFY can be found in Appendix A.

quickly without accumulating false positive localizations before being terminated. In contrast, the miss rate is minimized by a low n_{init} (to not miss any short tracks) and a high n_{term}, *e.g.* to bridge over difficult poses which cannot be correctly classified by the detectors. Trading off between miss rate and false positive rate, in terms of MOTA both a high n_{init} as well as a high n_{term} are beneficial.

The mismatch rate does not influence MOTA significantly due to the low number of mismatches. However, considered independently the mismatch rate is minimized by a high n_{init} and a high n_{term}. The latter is due to the inclusion of two types of

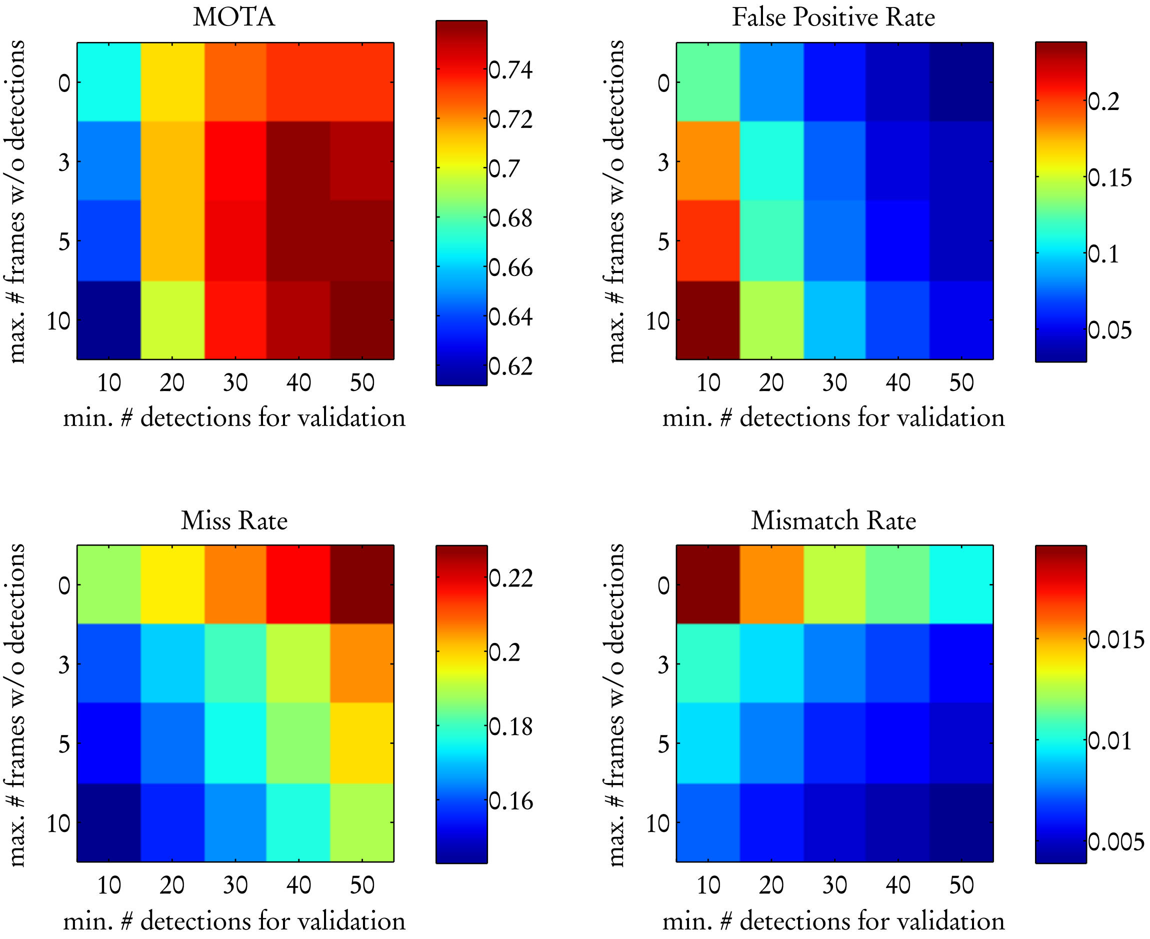


Figure 2.4.: In terms of MOTA, both a higher minimum number of detections for validation (n_{init}) as well as a high maximum number of frames before termination n_{term} are beneficial. These plots also visualize the trade-off between false positives, misses and mismatches as driven by initialization and termination. The false positive rate is low when many detections are required to validate a new track and tracks are quickly terminated. On the other hand, the miss rate is low when few detections are sufficient for track validation and tracks are continued for many frames without detections. The mismatch rate is high, when few detections are required for validation, but tracks are also quickly terminated. This leads to fragmented tracks, which are – if they stem from the same person – counted as mismatch in the MOTA metric.

max.	BBT Season 01 Episodes						Average
yaw	1	2	3	4	5	6	
0	48.5	37.7	40.4	41.0	49.8	40.6	43.0
15	53.1	40.4	43.8	44.5	53.4	43.4	46.4
30	52.3	40.9	43.4	44.8	52.4	43.6	46.2
45	52.1	41.0	42.8	46.1	55.0	44.0	46.8
60	54.3	43.5	43.9	45.4	54.1	44.4	47.6
90	56.9	44.3	46.4	47.2	54.8	48.0	49.6

Table 2.5.: Average track length in dependency of the maximum yaw angle of the detectors. Adding detectors for more poses does not only increase track recall, but also results in longer tracks, which is favorable for subsequent identification.

mismatches in the MOTA mismatch rate: i) true identity switches during a track (*bad* mismatches), and ii) assigning a new ID to a new tracklet which underlying person has been tracked before (*good* mismatches). Due to our occlusion handling, the number of good mismatches is usually higher than the number of bad mismatches (*e.g.*, due to short occlusions). In terms of MOTA, a high n_{term} is therefore beneficial, since it allows to bridge over short gaps and thus minimize the overall mismatch rate.

Minimum face size The MCT face detector always operates on the same fixed patch size due to the employed pyramid scanning. However, the performance of the detector on originally small image windows is usually worse than on larger windows. This can for example be explained by compression artifacts in the image which are – relative to the window size – stronger for smaller windows. Also, on a small scale, textures on natural objects are more likely to resemble a face. This consequently also influences tracking performance, since consistent false positive detections lead to false positive tracks (*cf*. Sec. 2.3, *Validation*). We therefore evaluate the influence of the minimum face size which we accept from the tracker (see Fig. 2.5). Below 30px, we observe more false positive than true positive tracks. Therefore, a rejection of tracks with a mean width < 30px increases overall MOTA.

Runtime The runtime of the tracker can be split into the following parts: initialization scan time t_{init}, time for resampling t_{re}, propagation t_{prop}, occlusion handling t_{occ} and for evaluation of the observation model t_{obs}. t_{init} and t_{obs} can be further divided into the time required to compute the underlying feature pyramid $t_{\text{pyr...}}$ and the evaluation of the detector cascades $t_{\text{det...}}$.

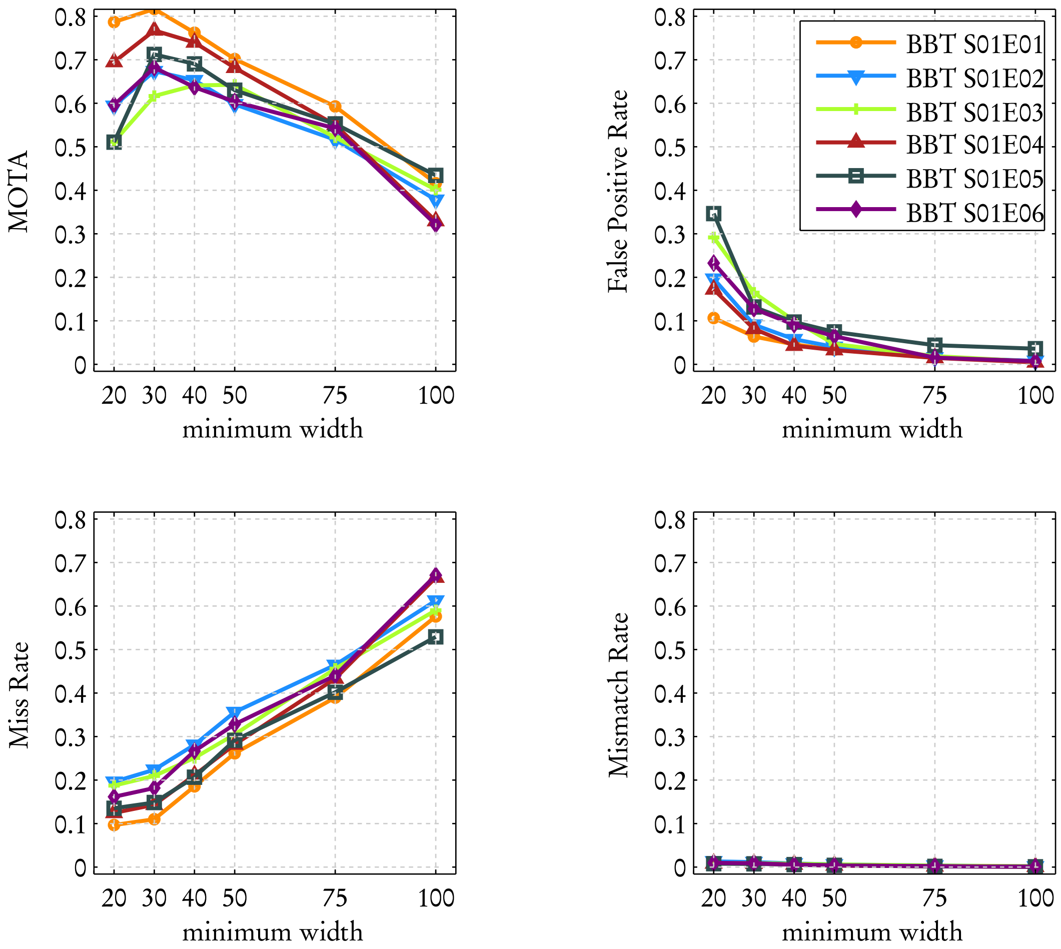


Figure 2.5.: Tracking performance over increasing minimum average track sizes. Rejecting tracks with an average width below 30px decreases the false positive rate faster than it increases the miss rate and thus leads to an improvement in MOTA.

We measure the timings on a video with resolution 1024 × 576px, using pyramid scale step 1.1 and the full bank of 47 detectors for initialization with $k = 5$, *i.e.* running the initialization scan every 5 frames. The measurements are performed single-threaded, *i.e.* no parallelization across multiple cores was performed for the purpose of this evaluation. Note that these timings are data dependent. For example, the run time of the detector depends on how many stages are completed for each detection window. However, they give an insight into the computational complexity of the different tracker components (see Tbl. 2.6).

The feature pyramid construction only has to be performed once per frame and can be shared among all detectors. In our implementation, we build separate feature pyramids

	#calls	time per call [ms]	total runtime [s]
t_{frame}	125	1564.9	195.6
t_{init}	25	7598.2	190.0
$\ldots t_{\text{pyrinit}}$	25	134.3	3.4
$\ldots t_{\text{detinit}}$	25	7463.9	186.6
t_{obs}	267	12.4	3.3
$\ldots t_{\text{pyrobs}}$	125	13.1	1.6
$\ldots t_{\text{detobs}}$	267	6.3	1.7
t_{re}	267	0.813	0.2
t_{prop}	267	7.760	2.1
t_{occ}	267	0.025	0.007

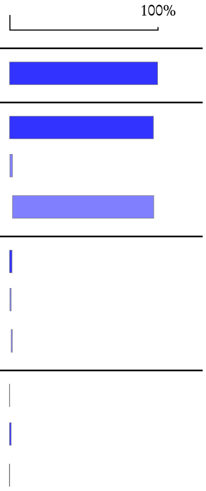


Table 2.6.: Average run times over a 5 second interval of a video with 25fps corresponding to a total of 125 frames. *#calls* denotes how often the corresponding component is called for the five seconds of the video. All track-specific components are called 267 times, since there are more than 2 tracks on average per frame.

for scanning and observation model, since the pyramid for the observation model can be restricted to a region of interest where particles are present and thus is faster to compute.

The average run time per frame t_{frame} is $1.56s$. Averaged across all frames, 97% of the time is spent during the initialization scan. The initialization is in turn mainly dominated by the evaluation of face detectors and roughly depends linearly on the number of employed detectors. To reduce the time required for initialization, we can (i) run only a subset of the detectors and (ii) increase k, the number of frame between to initialization scans. By running fewer detectors during initialization, we can trade off the coverage of pose classes and the run time required to run the detectors. Similarly, a higher k trades off the latency until a new face is detected by the tracker versus the run time. Finally, both the construction of the feature pyramid and the detector scans can easily be parallelized across multiple cores.

2.4.5. Comparison with other trackers

We compare the full-pose tracker with two other trackers from the literature for which results on the data set are available: the KLT-based tracker (Sivic et al., 2009) and an association-based face tracker (Roth et al., 2012) (ABT), which is an adaption of (Huang et al., 2008) to face tracking. We obtained predicted bounding boxes from both (Sivic

Approach	BUFFY Season 05 Episodes						Average
	1	2	3	4	5	6	
KLT (Sivic et al., 2009)	0.597	0.537	0.435	0.567	0.551	0.519	0.534
KLT (interpolated) (Sivic et al., 2009)	0.659	0.630	0.530	0.645	0.632	0.572	0.611
ABT (w/o high level) (Roth et al., 2012)		0.668					0.668
ABT (w/ high-level) (Roth et al., 2012)		0.719					0.719
Ours	0.750	0.707	0.656	0.715	0.721	0.706	0.709

Table 2.7.: MOTA rates for the episodes 1-6 of the 5th season of BUFFY. Our proposed tracker outperforms both the KLT-based approach as well as the low-level association-based approach (ABT). Both KLT and ABT approaches are offline approaches and require knowledge about the full video in advance, whereas we do not. With further high level information such as face identification scores, ABT is able to slightly outperform our approach on episode 2. However, this comes at the cost of a higher number of bad mismatches as can be seen in Tbl. 2.8.

et al., 2009) and (Roth et al., 2012) and use the same ground truth and evaluation pipeline to remove any influence from the evaluation procedure itself.

While the approach from Sivic et al. (2009) uses different detectors (a frontal and a profile detector), the association-based approach from Roth et al. (2012) is based on the same set of MCT detectors as our approach. Both other approaches are offline approaches, *i.e.* they require all frames to be available before tracking. They are therefore less suitable for online scenarios such as in a surveillance application. They also require to scan the full image with the full bank of detectors every frame, while we only scan every 5 frames, which gives our approach a significant run time advantage.

We report results in Tables 2.7 and 2.8. For the KLT-based tracker we report results for both the original tracker outputs, which only consists of the frames where one of the detectors fired, and an interpolated variant (denoted *KLT (interpolated)* in the tables), where we linearly interpolated between consecutive detections of a track which were more than one frame apart. The ABT approach also has two modes of operation, namely with and without a high-level stage of linking tracklets across larger frame gaps.

In terms of MOTA, our approach outperforms both the KLT-based and the ABT (w/o high level) approaches, while being on par with the ABT (w/ high level) approach. In Table 2.8, we compare the underlying failure rates between all three trackers. The approach from Sivic et al. (2009) suffers from a high miss rate, owing to the fact that only frontal and full profile detectors are used. Using a higher number of intermediate pose

Approach	BUFFY Season 05 Episode 02					
	MOTA	MR	FPR	BMM	GMM	TR
KLT (Sivic et al., 2009)	0.537	0.424	0.024	1 (0.000)	105 (0.015)	0.671
KLT (interpolated) (Sivic et al., 2009)	0.630	0.292	0.063	2 (0.000)	107 (0.015)	0.682
ABT (w/o high level) (Roth et al., 2012)	0.668	0.207	0.062	3 (0.000)	451 (0.063)	0.851
ABT (w/ high-level) (Roth et al., 2012)	0.719	0.168	0.097	62 (0.009)	79 (0.011)	0.851
Ours	0.707	0.147	0.117	6 (0.001)	211 (0.029)	0.885

Table 2.8.: Detailed evaluation scores for BUFFY S05E02. Our approach's favorable performance is due to a low miss rate (MR), also reflected in the highest track recall (TR) of all approaches. While ABT with high level linkage outperforms our approach by about 0.01 in MOTA, it comes at the cost of a high number of bad mismatches (BMM). These are especially undesired for further identification since each BMM links two different persons within the same track.

classes suggests to be a good way to reduce the number of missed faces. The approach by Roth et al. (2012) also exhibits a higher miss rate than our tracker, possibly due to a more aggressive filtering of false positive tracklets, but also a notably high number of (albeit good) mismatches. The inclusion of the high-level association (*ABT w/ high level*) reduces the number of good mismatches, but also introduces a high number of bad mismatches, *i.e.* track switches. Finally, our approach has the lowest miss rate of all three trackers, which is also reflected in the track recall (see Tbl. 2.7).

Chapter 3

Semi-supervised Learning with Constraints

In the previous chapter, we localized faces in video data. Given the tracks we obtained from the tracker, we are now interested in identifying each of them, *i.e.* assigning a unique identity or name to each of them. In the context of realistic multimedia data, identification can be seen as an *open-set* problem, *i.e.* there are usually background persons (*e.g.*, extras) which are not important to the story and instead can be collectively rejected as *unknowns*.

The nowadays prevalent way for face recognition is to use a supervised machine-learning approach. Training data for each person is collected and a multi-class classifier is trained. For open-set recognition, we further have to decide whether a person is actually among the known persons, for example based on the classifiers confidences. This machine learning-based approach to face recognition can give good results when accurate and enough training data is available.

However, obtaining manually labeled training data is a tedious, time-consuming and thus expensive task. We will therefore follow Everingham et al. (2006)'s approach and make use of available subtitles and transcripts to obtain labels for face tracks automatically. Further, we assume that all face tracks are available at training time, which is often the case in the context of multimedia data. For example, if the goal is to display additional information on characters during the playback of a TV episode, the identification can be performed once – offline – beforehand. Since even unlabeled tracks contain information, *e.g.* about the distribution of possible face appearances, it can be beneficial to make use of them during training. If both labeled and unlabeled data is used during training, the learning is called to be *semi-supervised*. If *all test* data is available during training, the learning problem can be approached via *transduction*, *i.e.* transferring the given labels to the test data, in contrast to being *inductive*, *i.e.* learning a generalizing model first from the training data and then inferring the labels for the test data from the model. Whether transductive learning is an easier task, since labels only have to be propagated from the labeled to the unlabeled examples without the need to learn a generalizing model first, is a topic of ongoing discussion in the machine-learning community (Chapelle, Olivier Schölkopf, Bernhard Zien, 2006). In this chapter, we approach the problem from the semi-supervised/inductive perspective because our employed model indeed has generalizing capabilities and as such can also classify unknown test data.

Finally, natural constraints arise within and between tracks and can be determined automatically. For example, all frames in a track are assumed to show the same person. This results in a *positive* or *must-link* constraint, *i.e.* we can model already during training that such samples are to be assigned the same identity. For this, it is important that the tracker exhibits a low number of bad mismatches, as we analyzed in the previous chapter. Similarly, we can also obtain *negative* or *cannot-link* constraints from the data. Two tracks which co-occur in the same frame cannot stem from the same person and thus should never be assigned the same identity [1]. This induces not only constraints between samples in the same frame, but between all pairs of frames of the two tracks, again due to the previous assumption that all frames in a track belong to the same person. This is a notable difference to single images (*e.g.*, from a news page), where such constraints would only be valid for detections within the one image.

In this chapter, we approach the problem of person identification as a semi-supervised learning problem with constraints. The goal is to automatically identify all face tracks by training discriminative multi-class classifiers from automatically obtained, weakly-supervised track labels, additional unlabeled data and automatically generated constraints between tracks. We integrate all three sources of information in a common learning framework.

[1] This is of course not true in the presence of mirrors in the scene. There is one such case in one video of our data set, but compared to the amount of true negative constraints between tracks this occurrence is negligible.

3.1. Background and related work

The literature on face recognition is vast. We will therefore focus the discussion – after a very brief introduction of supervised face recognition – on related semi-supervised learning and clustering approaches. For a review of face recognition in general, we refer the reader to surveys by Zhao and Chellappa (2003) and Jafri and Arabnia (2009).

Supervised face recognition For machine-learning-based *supervised* face recognition, the goal is usually to learn a predictor $\mathscr{F} : \mathscr{X} \rightarrow \mathscr{Y}$, using a set of training samples $\mathscr{X}_l = \{(x_i, y_i)\}_{i=1}^{L}$ with descriptors $\mathbf{x}_i$ and labels y_i. During testing, labels of all test samples $\{\mathbf{x}_j\}_{j=L+1}^{L+T}$ are inferred using $\mathscr{F}$. Descriptors $\mathbf{x}_i$ are usually obtained by first aligning the face to a canonical pose, and the performing a feature extraction, encoding and possibly dimensionality reduction to obtain a compact face descriptor.

Many different choices for $\mathscr{F}$ are described in the literature, for example Support Vector Machines (*e.g.*, Berg and Belhumeur (2012); Heisele et al. (2001)) and AdaBoost (*e.g.*, Zhang et al. (2004)), but also non-parametric approaches such as k-Nearest-Neighbors (Ahonen et al., 2004; Fischer et al., 2012). Similarly, a multitude of alignment methods (*e.g.*, Berg and Belhumeur (2012)) and descriptor extraction methods (*e.g.*, Ahonen et al. (2004); Ekenel and Stiefelhagen (2006); Taigman et al. (2014)) have been proposed.

In the context of multimedia data, face recognition has been also explored in supervised settings. For example, Ortiz et al. (2013) exploits the Public Figures data set as training set. Also, different approaches which obtain labels automatically from associated sources operate in a supervised setting. Berg et al. (2004) obtain labels from image captions, while Everingham et al. (2006), Sivic et al. (2009) and Bojanowski et al. (2013) from transcripts and subtitles. As classifiers, nearest neighbor (Everingham et al., 2006) and multiple kernel learning (Sivic et al., 2009) have been explored, amongst many more customized classifiers, for example to handle ambiguous labels (Berg et al., 2004; Cour et al., 2009) or jointly classify multiple modalities (Bojanowski et al., 2013).

Semi-supervised learning Semi-supervised learning differs from supervised learning in that additional *unlabeled* training samples $\mathscr{X}_u = \{\mathbf{x}_i\}_{i=L+1}^{L+U}$ are available at training time. The labels for these are not known, but the assumption is that they provide additional information about the distribution of the underlying structure of the data (*e.g.*, the "face-manifold") and thus can help to build a better classifier.

One common assumption about the distribution of data is that samples of the same class form clusters in the descriptor space (the *cluster assumption*). To exploit that, approaches for example encourage decision boundaries in low density regions between classes. By maximizing the margin of hyperplanes on both labeled and unlabeled data Xu et al. (2005) and Joachims (1999) extend the Support Vector Machine principle to semi-supervised and transductive learning, using a symmetric hinge loss on the unlabeled data. Smooth and ramped variations of the hinge loss have been explored by Chapelle and Zien (2004) and Collobert et al. (2006), respectively. As another extension to supervised classifiers, Grandvalet and Bengio (2005) proposed entropy-based regularization of the classifiers' model parameters. Closely related is also the idea of self learning and Expectation Maximization-based approaches, where unlabeled examples are assigned intermediate labels using the current classifier's model. Those intermediate labels are then used to again update the model parameters. Variants of these ideas have been employed for recognizing faces in images (*e.g.*, Zhao et al. (2011)) or web videos (*e.g.*, Rim et al. (2011)), and can also be used for generative models (Nigam et al., 2000). If different descriptors for the same data are available, for example face and clothing descriptors, one can iteratively *co-train* two classifiers which provide labels for the unlabeled samples of the respective other class (Blum and Mitchell, 1998).

The iterative nature of self-learning and EM-style approaches is usually due to a non-convex part of the loss function. To avoid local minima, convex relaxations to originally non-convex loss functions have been proposed (Joulin and Bach, 2012). Xu et al. (2005) formulate the problem of finding maximum margin hyperplanes as a convex integer problem.

Unlabeled data can also be used in the context of multiple instance learning. Joulin and Bach (2012) employ an entropy penalty to encourage a uniform class distribution in a bag. To incorporate unlabeled data within multiple instance boosting, Zeisl et al. (2010) employ a cross-entropy loss between a prior belief of class label distribution and the obtained distribution on the unlabeled data. A variant thereof was applied to face recognition in TV series using ambiguous labels obtained from subtitles and transcripts (Köstinger et al., 2011).

In the context of labeling person identities in photos, different approaches using Markov random fields (Anguelov et al., 2007; Gallagher and Chen, 2007; Lin et al., 2010) have been explored. Unlabeled examples are incorporated via pair-wise relations to other labeled and unlabeled examples. In contrast to earlier discussed approaches, a joint probabilistic estimation of identity labels can be performed directly, instead of learning a generalizing model first. Gallagher and Chen (2008a) exploit unlabeled data via co-segmentation, *i.e.* segmenting the upper body regions of multiple instances jointly, in order to obtain better clothing models for subsequent identification.

Learning with constraints A different way utilizing unlabeled data is through pairwise relations between them. In the context of images and videos such constraints arise for example from tracking, co-occurrence reasoning or user input.

One way to exploit such constraints is through an additional loss term on unlabeled data to a standard supervised loss. For example, Melacci et al. (2009) integrate constraints into kernel ridge regression by penalizing constraint violations on unlabeled data. Yan et al. (2006) employ a similar idea and enhance kernel logistic regression with a constraint loss for person identification in a camera network. Both approaches incorporate unlabeled data not directly, but through the pairwise constraints, in contrast to the previously discussed semi-supervised approaches.

Pairwise constraints essentially encode the information which descriptors should obtain the same label, and which should not. From the perspective of distances, this means that descriptors with a must-link constraint should have a low distance, and with a cannot-link should have a high distance. To exploit this, different approaches have been proposed to learn a specialized distance metric, for example for image retrieval (Hoi et al., 2006) or face recognition (Guillaumin et al., 2012). Cinbis et al. (2011) make use of must-link and cannot-link constraints in order to learn a face- and cast-specific metric in order to improve face clustering and identification in TV series. However, they rely on supervised labeling of clusters in order to perform the actual identification.

In the context of clustering, must-link and cannot-link constraints are often employed as additional cues which clusters to join or keep separated, and as such constrained variants of k-means (Bilenko et al., 2004), maximum margin clustering (Zeng and Cheung, 2012) or spectral clustering have been proposed (Li and Liu, 2009). For simultaneous tracking and clustering of faces, Wu et al. (2013b) embed pairwise constraints between face tracks into a Hidden Markov Random Field-based clustering approach. When clustering faces, the resulting clusters are usually not labeled with unique labels. However, they can be assigned unique identity labels by a separate mechanism, *e.g.*, by a separately trained multi-class SVM (Yu et al., 2011) or user input (Ramanan et al., 2007b).

Markov random field-based approaches for face recognition usually model pairwise relations between faces and therefore constraints naturally integrate in such models. For example, Anguelov et al. (2007) integrate a uniqueness constraint, enforcing that two persons in the image are not identified with the same identity. However, the constraint is not strictly enforced, as they found empirically that allowing a small number of conflicting identity assignments is beneficial for overall performance. Instead of constraints, Lin et al. (2010) model a co-occurrence likelihood in a pairwise potential, *i.e.* people that co-occur often are also more likely to be jointly assigned to two faces in other images.

3.1.1. Discussion and contribution

There are many examples in the literature how adjoining data can be exploited for automatic generation of labels. The method of Everingham et al. (2006) of aligning subtitles (*what* is spoken *when*) and transcripts (*who* speaks *what*) is especially compelling, since subtitles and transcripts are ubiquitously available for most popular TV series. A drawback of this method is that the obtained labels are not pure due to the inherent visual speaker detection step. Nevertheless, we will employ a similar method, since it works very well in practice and allows for an automatic labeling of training data. We will analyze to which extend the non-purity of the labels affects results.

The literature on semi-supervised learning clearly motivates the use of unlabeled data for training better models without additional supervision. There is an abundance of unlabeled data available in the multimedia context, and different approaches have shown improvements on recognizing faces and characters by exploiting data with no labels. The cluster assumption leads to different variants of the same idea, and eventually most result in a large-margin-encouraging addition to the respective original supervised loss of different classifiers. In contrast to transductive approaches, a generalizing classifier has the advantage of being able to further classify new unseen test data. This is beneficial when we obtain a new episode of a TV series for which we already have trained models for all cast members (and do not want to re-run the full learning procedure every time). We will therefore also start out with a supervised classifier. Following unlabeled large-margin ideas, we employ an entropy-based term in the loss function to incorporate unlabeled data, similar to Grandvalet and Bengio (2005).

Finally, constraints between faces and face tracks arise naturally in videos. For clustering, it is easily perceivable that they can help to link clusters (*e.g.*, across a pose change) or to avoid errors. Also in supervised settings, constraints have been shown to help in training better models. However, in these approaches, unlabeled data was only considered in the context of constraints, not by itself as in the semi-supervised literature. If an unlabeled sample is not present in a constraint, it would not be used for learning.

In this chapter, we consider all resources together. We learn person models from automatically obtained labels, unlabeled data and constraints in a common framework. We incorporate these three sources of information in a common loss function for training a multi-class classifier (Sec. 3.2). We apply the proposed learning framework to the task of character naming in TV series and achieve state-of-the-art results (Sec. 3.3).

3.2. Semi-supervised learning with constraints

Let $\mathcal{X}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^L$ denote training data $\mathbf{x}_i$ with associated labels $y_i \in \mathcal{Y}$. The problem of character naming is inherently a multi-class problem, thus $|\mathcal{Y}| = K$ and, without loss of generality, we assume $\mathcal{Y} = \{1, \dots, K\}$. We further have additional unlabeled data $\mathcal{X}_u = \{\mathbf{x}_i\}_{i=L+1}^{L+U}$. Positive and negative constraints between data points $\mathcal{C} = \{(\mathbf{x}_{i1}, \mathbf{x}_{i2}, c_i)\}_{i=1}^C$, where $c_i \in \{+1, -1\}$, denote pairs of features which belong to the same character and should be assigned the same identity ($c = +1$), or do not belong to the same character and should not be assigned the same identity ($c = -1$).

Using the given training data, we are interested in learning the set of parameters θ of a classifier, which maps a new descriptor to one of the K classes

$$\mathcal{F}_\theta(\mathbf{x}) : \mathcal{X} \to \mathcal{Y} \quad . \tag{3.1}$$

A common way to learn θ from training data is to define a loss function over the training data. By minimization of the loss, we obtain the parameter set θ^* that best fits the given training data

$$\theta^* = \operatorname*{argmin}_\theta \mathcal{L}(y|\mathcal{X}_l; \theta) \quad . \tag{3.2}$$

Different choices of $\mathcal{F}$ lead to different types of classifiers. The definition of $\mathcal{L}$ determines the way in which the parameters θ of the classifiers are learned. For a supervised classifier, the loss function for example only takes into account labeled examples $\mathcal{X}_l$.

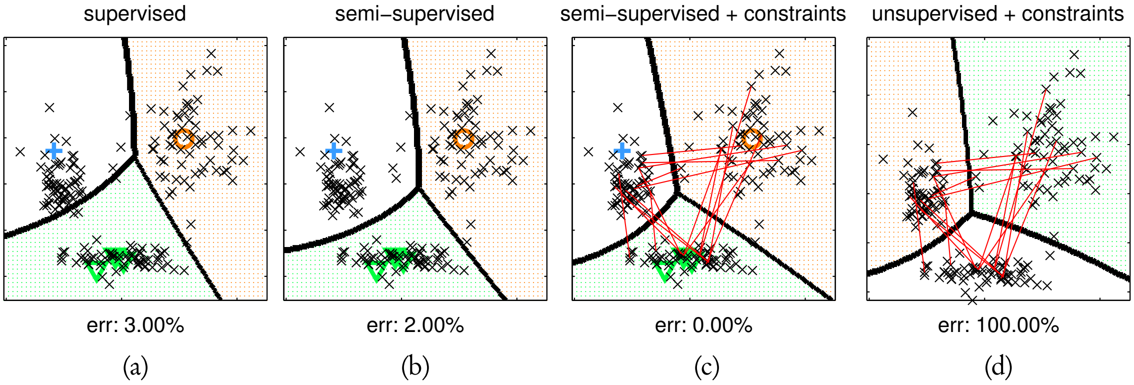


Figure 3.1.: Visualization of the effect of the different terms of the loss function on a toy example. The denoted error is the joint error on labeled *and* unlabeled data.
(a) $\mathscr{L}_l$: Supervised learning from labeled data (colored data points $+/\bigcirc/\nabla$) only.
(b) $\mathscr{L}_l + \mathscr{L}_u$: Semi-supervised learning by additionally taking unlabeled data (black $\times$) into account. The decision boundaries are encouraged to better fit the underlying distribution.
(c) $\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$: By further taking into account negative constraints between data points, the error reduces to 0.
(d) $\mathscr{L}_u + \mathscr{L}_c$: Even without using the labels, it is possible to still find meaningful structure in the data using the entropy and constraint loss. However, the assignment to the classes is not uniquely defined and has to be done in a separate step.

Previous work extended supervised loss functions by addition terms to include unlabeled data (*e.g.*, Grandvalet and Bengio (2005)) or constraints on unlabeled data (*e.g.*, Yan et al. (2006)). We follow this approach and define a combined loss function that takes into account (i) labeled data $\mathscr{X}_l$, (ii) unlabeled data $\mathscr{X}_u$ and (iii) constraints $\mathscr{C}$:

$$\begin{aligned} \mathscr{L}(\mathscr{X};\theta) &= \mathscr{L}(y_l, y_c; \mathscr{X}_l, \mathscr{X}_u, \mathscr{C}, \theta) && (3.3) \\ &= \mathscr{L}_l(y_l; \mathscr{X}_l, \theta) + \mathscr{L}_u(\mathscr{X}_u, \theta) + \mathscr{L}_c(y_c; \mathscr{C}, \theta). && (3.4) \end{aligned}$$

In this generic form, $\mathscr{L}_l$ denotes the supervised loss on labeled data, $\mathscr{L}_u$ the loss term on unlabeled data and $\mathscr{L}_c$ the loss over the given constraints.

We will now first introduce our model for $\mathscr{F}$ and then describe the different terms of the loss function in more detail. The influence of different parts of the loss function are visualized in Fig. 3.1 on a toy example.

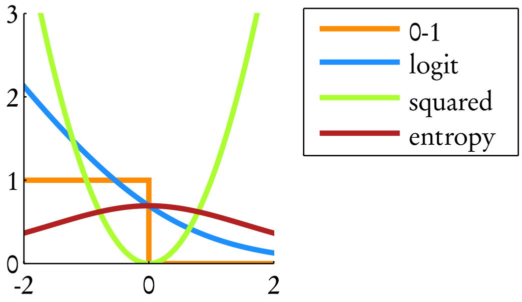


Figure 3.2.: Visualizations of the employed loss functions.

3.2.1. Model

Multinomial logistic regression (MLR) (Hastie et al., 2009) belongs to the family of log-linear models and is a classical choice for multi-class classification. One of the advantages of MLR is that its results can directly be interpreted as probabilities of a data point belonging to class k with

$$P(y=k|\mathbf{x};\theta)=\frac{e^{\theta_k^T\mathbf{x}}}{\sum_z e^{\theta_z^T\mathbf{x}}} \tag{3.5}$$

with $P(y=k|\mathbf{x};\theta)\in[0,1]$ and $\sum_k P(y=k|\mathbf{x};\theta)=1$. The model is defined by parameter vectors θ_k, one for each class. The full parameter set is given by $\theta=[\theta_1,\cdots,\theta_K]$. Due to the constraint $\sum_k P(y=k|\mathbf{x};\theta)=1$, there are only $K-1$ free parameter vectors and consequently the parameter vector θ_K is usually fixed as 0.

To classify a sample $\mathbf{x}$ under this model, we compute the most likely class as

$$\mathscr{F}_\theta(\mathbf{x})=\operatorname*{argmax}_k P(y=k|\mathbf{x};\theta)=\operatorname*{argmax}_k \frac{e^{\theta_k^T\mathbf{x}}}{\sum_z e^{\theta_z^T\mathbf{x}}} \quad . \tag{3.6}$$

Due to the argmax and the monotony of e^x, we can simplify the classification rule to

$$\mathscr{F}_\theta(\mathbf{x})=\operatorname*{argmax}_k \ \theta_k^T\mathbf{x} \quad . \tag{3.7}$$

For notational brevity, we denote $P_\theta^k(\mathbf{x}):=P(y=k|\mathbf{x};\theta)$ in the rest of this chapter.

Kernelization Multinomial logistic regression can be extended to non-linear decision boundaries by expanding $\mathbf{x}$ by a feature map function $\Phi(\mathbf{x})$. $\mathbf{x}$ can be expanded explicitly, however, for high dimensional or even infinite dimensional mappings this is often infeasible in practice. Instead of computing $\Phi(\mathbf{x})$ directly, $\theta_k^T\mathbf{x}$ can be replaced by a function $f(\mathbf{x})$. According to the representer theorem (Kimeldorf and Wahba, 1971) $f(\mathbf{x})$ has the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} \theta_{ki} K(\mathbf{x}, \mathbf{x}_i) \quad , \tag{3.8}$$

where $K(\cdot,\cdot)$ is a positive definite reproducing kernel.

3.2.2. Supervised loss

In order to learn the parameter set θ from labeled training samples $\mathscr{X}_l$, we use the standard negative log-likelihood as loss

$$\mathscr{L}_l(y_l; \mathscr{X}_l, \theta) = -\frac{1}{L} \sum_{i=1}^{L} \sum_{k=1}^{K} \mathbf{1}[y_i{=}k] \ln(P_\theta^k(\mathbf{x}_i)) + \lambda ||\theta||^2 \tag{3.9}$$

where $\mathbf{1}[\cdot]$ is the indicator function. The regularization term $\lambda||\theta||^2$ corresponds to a zero-mean Gaussian prior on the parameters. Its purpose is to prevent overfitting on the training data and its influence is controlled by the hyper-parameter λ.

This loss is convex and can be efficiently minimized with standard gradient descent techniques. The gradient of Eq. 3.9 with respect to θ is

$$\frac{\partial}{\partial \theta_k} \mathscr{L}_l = 2\lambda\theta_k - \frac{1}{L} \sum_{i=1}^{L} \mathbf{x}_i \cdot \left(\mathbf{1}[y_i{=}k] - P_\theta^k(\mathbf{x}_i)\right) \quad . \tag{3.10}$$

3.2.3. Entropy loss for unlabeled data

While the unlabeled data $\mathscr{X}_u$ does not carry information about its class membership, it can be informative about the distribution of data points in regions without labels. As discussed in the introduction and discussion of related work, the decision boundaries should also respect the distribution of unlabeled data. That is, the decision boundaries should preferably lie in low-density regions (see the toy example in Fig. 3.1 for a visual example).

A common way to achieve this is to include an entropy term into the loss function in order to encourage uniformly distributed class membership across the unlabeled data (*e.g.*, Köstinger et al. (2011); Zeisl et al. (2010)). Instead, we use the entropy function as a penalty on having the decision boundaries close to unlabeled data points (see Fig. 3.2)

$$h(\mathbf{x}_i) = -\sum_{j=1}^{K} P_\theta^j(\mathbf{x}_i) \ln(P_\theta^j(\mathbf{x}_i)) \quad . \tag{3.11}$$

In order to compute the loss on $\mathcal{X}_u$, we sum over all unlabeled data points

$$\begin{aligned} \mathcal{L}_u(\mathcal{X}_u;\theta) &= \frac{\mu}{M}\sum_{i=1}^{M} h(\mathbf{x}_i) \qquad (3.12)\\ &= -\frac{\mu}{M}\sum_{i=1}^{M}\sum_{j=1}^{K} P_\theta^j(\mathbf{x}_i)\ln(P_\theta^j(\mathbf{x}_i)) \quad , \end{aligned}$$

where μ controls the relative influence of the unlabeled data on the total loss. For our model of P this leads to the following gradient:

$$\begin{aligned} \frac{\partial}{\partial \theta_k}\mathcal{L}_u &= -\frac{\mu}{M}\sum_{i=1}^{M}\frac{\partial}{\partial \theta_k}\left[\sum_{j=1}^{K} P_\theta^j(\mathbf{x}_i)\ln(P_\theta^j(\mathbf{x}_i))\right] \\ &= -\frac{\mu}{M}\sum_{i=1}^{M}\mathbf{x}_i P_\theta^k(\mathbf{x}_i)\cdot\left[\sum_{j=1}^{K}\left(\mathbf{1}[k=j]-P_\theta^j(\mathbf{x}_i)\right)\left(1+\ln(P_\theta^j(\mathbf{x}_i))\right)\right] \quad . \end{aligned} \tag{3.13}$$

3.2.4. Constraints

Finally, we include pair-wise constraints between training samples $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$. The constraint $(\mathbf{x}_{i1}, \mathbf{x}_{i2}, c_i)$ specifies whether $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$ belong to the same class ($c_i = 1$) or not ($c_i = -1$). Such constraints arise for example from temporal relations between face tracks, *i.e.*, two tracks which temporally overlap cannot belong to the same person, and can be automatically generated without manual effort. Note that, in general, the class memberships of both $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$ are unknown. Nevertheless, constraints are not only important between unlabeled data points, but also between pairs of unlabeled and labeled data points, and even between two labeled data points. Since we plan to obtain labels in an automated manner, they can contain errors, which in turn can potentially be corrected by a constraint during training.

Intuitively, for a negative constraint the product of the likelihood of features $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$ belonging to different classes

$$P(y_{i1} \neq y_{i2}) = \sum_{j=1}^{K} \sum_{\substack{l=1 \\ l \neq j}}^{K} P_\theta^j(\mathbf{x}_{i1}) P_\theta^l(\mathbf{x}_{i2}) \tag{3.14}$$

should be high. Since the $P_\theta^k(\cdot)$ sum up to one, we can simplify the above to

$$P(y_{i1} \neq y_{i2}) = 1 - \sum_{j=1}^{K} P_\theta^j(\mathbf{x}_{i1}) P_\theta^j(\mathbf{x}_{i2}) \quad . \tag{3.15}$$

We use the negative log-likelihood of the features belonging to different classes as loss

$$\begin{aligned} \mathscr{L}_c(c_i; \mathscr{C}, \theta) &= -\frac{\gamma}{L} \sum_{i=1}^{L} \ln(P(y_{i1} \neq y_{i2})) \\ &= -\frac{\gamma}{L} \sum_{i=1}^{L} \ln \left(1 - \sum_{j=1}^{K} P_\theta^j(\mathbf{x}_{i1}) P_\theta^j(\mathbf{x}_{i2}) \right) , \end{aligned} \tag{3.16}$$

where γ controls the relative influence of the constraint loss. The derivative of $\mathscr{L}_c$ with respect to θ_k is

$$\begin{aligned} \frac{\partial}{\partial \theta_k} \mathscr{L}_c &= -\frac{\gamma}{L} \sum_{i=1}^{L} \frac{\partial}{\partial \theta_k} \ln(P(y_{i1} \neq y_{i2})) \\ &= -\frac{\gamma}{L} \sum_{i=1}^{L} \left[\frac{-1}{P(y_{i1} \neq y_{i2})} \frac{\partial}{\partial \theta_k} \sum_{j=1}^{K} P_\theta^j(\mathbf{x}_{i1}) P_\theta^j(\mathbf{x}_{i2}) \right] \\ &= \frac{\gamma}{L} \sum_{i=1}^{L} \frac{1}{P(y_{i1} \neq y_{i2})} \left[\left(\mathbf{x}_{i1} + \mathbf{x}_{i2} \right) P_\theta^k(\mathbf{x}_{i1}) P_\theta^k(\mathbf{x}_{i2}) \right. \\ &\qquad \left. - \left(\mathbf{x}_{i1} P_\theta^k(\mathbf{x}_{i1}) + \mathbf{x}_{i2} P_\theta^k(\mathbf{x}_{i2}) \right) P(y_{i1} = y_{i2}) \right] . \end{aligned} \tag{3.17}$$

3.2.5. Minimization of the loss

We first collect training data from all available episodes, and train one joint multi-class classifier from supervised data, unsupervised data and constraints by minimization of the joint loss function (Eq. 3.4) via L-BFGS (Liu and Nocedal, 1989), a limited memory variant of the Broyden-Fletcher-Goldfarb-Shanno algorithm. Using L-BFGS for the minimization requires the gradient of the loss function to be available, which we have given together with the respective losses.

Taking into account all available training data from multiple episodes at the same time is unfortunately computationally infeasible, especially for the kernelized version of the multinomial logistic regression. We therefore reduce the data by subsampling, effectively removing features that were temporally nearby and therefore presumably visually similar. For the kernel computation we further randomly select prototypes instead of working with the full kernel matrix similar to (Lee and Mangasarian, 2001).

3.3. Automatic character naming

We apply the proposed learning framework to the task of character naming in videos. In this chapter, we only consider *face* tracks for identification similar to Everingham et al. (2006), Köstinger et al. (2011) and Sivic et al. (2009) as obtained from our tracker from Chapter 2. We will consider clothing and other modalities in the next chapter, where we will use the obtained track identities from this chapter as one input modality.

3.3.1. Data set

We evaluate our approach on 6 episodes each of season 1 of *The Big Bang Theory* (BBT-1 to BBT-6) and season 5 of *Buffy the Vampire Slayer* (BF-1 to BF-6). We employ the face tracks obtained from our tracker as described in Chapter 2. Since our approach does not deal with false positives explicitly, we remove false positive tracks manually to avoid a distracting influence on the recognition performance. An automatic method to detect false positive tracks is described in (Tapaswi et al., 2014c).

The data set consists of a total 3921 face tracks for BBT, and 5861 face tracks for BUFFY. This corresponds to a track recall of about 80% for BBT (*cf*. Table 2.3) and 90% for BUFFY (*cf*. Table 2.4). For an overview over some statistics of the data set see Table 3.1.

As discussed already for the tracking, the two series differ in their filming style and therefore pose different challenges. Most notably for *identification* is the difference in number of characters. BBT consist of a main cast of 5 people with 1 to 3 supporting

	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6
# characters	6	5	7	8	6	6	12	13	14	15	15	18
# face tracks	658	615	660	613	524	851	796	1004	1194	900	840	1127
# unknown tracks	9	2	98	45	61	200	10	138	9	48	107	71
# speaking tracks	147	104	132	126	78	116	158	192	177	186	174	211
speaker precision	89.12	87.50	93.18	88.89	92.31	87.93	89.24	82.81	81.36	87.10	88.51	86.73
speaker recall	19.91	14.80	18.64	18.27	13.74	11.99	17.71	15.84	12.06	18.00	18.33	16.24

Table 3.1.: Overview over basic statistics of the data set and the speaker assignment performance on BBT and BUFFY. For both series, on average 16% of all face tracks are assigned an identity by the subtitle-transcript alignment and speaker detection. The average precision for BBT is 90% and for BUFFY 86%.

characters per episode, while BUFFY has a main cast size of around 12 characters and in specific episodes up to 18 important characters. We denote every character as *known* and assign them a unique identity, if they are named in the plot or have a significant role and a sufficient number of tracks. In addition, both series contain unnamed extras, which we denote as *unknowns*.

BBT's plot takes mostly place in well-lit indoor settings, while BUFFY has many outdoors scenes with poor lighting conditions. On the other hand, BUFFY contains a sizable number of close-up shots, resulting in a long tail in the distribution of face sizes (*cf*. Fig. 2.2).

Tables 3.2 and 3.3 show in their second column each character's number of face tracks accumulated over the six episodes of BBT and BUFFY, respectively. The number of face tracks varies between over 1000 tracks for the main character and less than 10 for some supporting characters. We label each of the face tracks with ground truth identities in order to perform a quantitative evaluation of the identification approach.

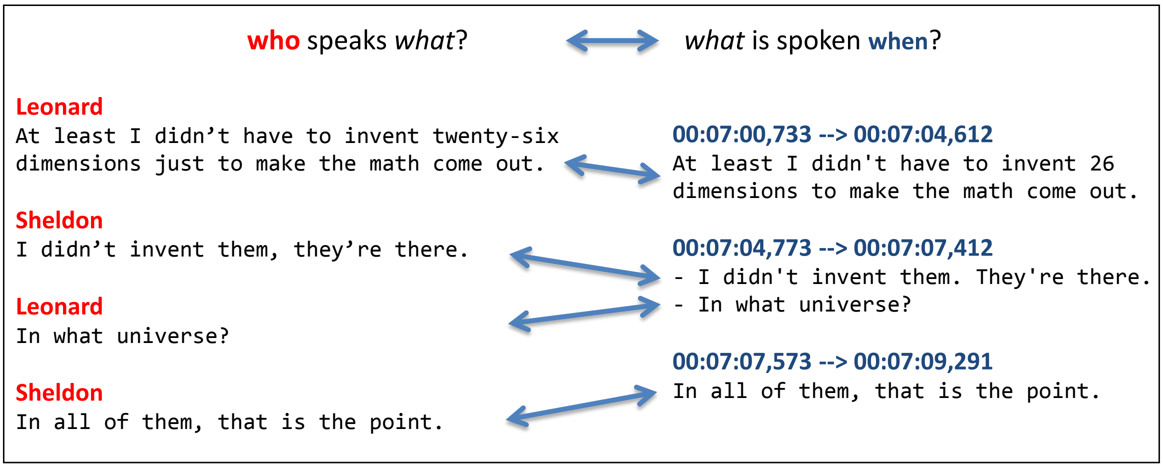
3.3.2. Preprocessing

The preprocessing of the data for character identification consists of multiple steps. We first detect shot boundaries and track faces as discussed in Chapter 2. As a second step, we estimate facial landmarks in every frame of each track, which are subsequently used in both speaker detection and face alignment. Keeping in mind the large amount of multimedia data, we are especially interested in an identification scheme that does not require manual supervision. We therefore follow Everingham et al. (2006) and align transcripts to subtitles, and perform visual speaker detection to determine the current speaking track. In this way, we obtain labels for some of the face tracks in an automatic manner. We will briefly discuss the process of facial landmark estimation, subtitle-transcript alignment and speaker detection in the following.

Facial landmark detection We estimate the location of facial landmarks using the Supervised Descent Method (SDM) by Xiong and la Torre (2013) which is a state-of-the-art method for facial landmark detection. SDM starts with a rough estimate of the landmark positions, *e.g.*, we use the weighted mean landmark positions relative to the tracker's detections in the current frame. Iteratively, the landmark positions are refined with a cascade of update steps, where each update vector is determined via joint linear regression based on SIFT features extracted around the current landmark positions. Multiple regression matrices are learned on training data for each of the update steps. The procedures converges after as few as four update steps and requires in total around 30ms per face.

Weak labels from speaking faces Following Everingham et al. (2006), we align subtitles with transcripts from the web in order to combine the timing component of subtitles (*what* is spoken *when*, but usually without speaker identity) with the identities from the transcripts (*who* speaks *what*, but without timing information). Using the common text to align the two information sources, we can obtain *who* is speaking *when*, *i.e.* we obtain labeled intervals (s_i, e_i, y_i) with start time s_i, end time e_i and associated speaker identity y_i. In order to associate the speaker identity information with a face track, we further estimate visually which of the co-occurring faces is speaking during the given time interval. The visual speaker detection is important to assign the identity to only one face track if multiple tracks are present. While other approaches do without speaker detection and handle possible ambiguities during training (*e.g.*, Cour et al. (2010)), the unique assignment of labels to the one speaking face allows for a less complex learning approach.

Subtitle-transcript alignment We align transcripts and subtitles on word level, *i.e.* we regard both texts as long sequences of words. We define the *best* alignment as the one requiring the minimum number of word operations (word insertions, deletions, and replacements) to transform one sequence into the other. This is closely related to computing the Levenshtein distance (Levenshtein, 1966) between two strings, albeit



who speaks *what*? ⟷ *what* is spoken **when**?

Leonard
```
At least I didn’t have to invent twenty-six
dimensions just to make the math come out.
```

Sheldon
```
I didn’t invent them, they’re there.
```

Leonard
```
In what universe?
```

Sheldon
```
In all of them, that is the point.
```

00:07:00,733 --> 00:07:04,612
```
At least I didn't have to invent 26
dimensions to make the math come out.
```

00:07:04,773 --> 00:07:07,412
```
- I didn't invent them. They're there.
- In what universe?
```

00:07:07,573 --> 00:07:09,291
```
In all of them, that is the point.
```

Figure 3.3.: Matching of transcripts to subtitles to determine *who* is speaking *when*.

using a different cost function. We compute the best alignment between word sequences s and t implicitly using the following recursive cost function

$$
\mathrm{d}_{s,t}(i,j) = \min \begin{cases} d_{s,t}(i-1,j-1)1_{\{s(i)\neq t(j)\}} \cdot \min(3,\ \mathit{leven}(s(i),t(j))) \\ d_{s,t}(i-1,j) + \mathit{len}(s(i)) \\ d_{s,t}(i,j-1) + \mathit{len}(t(j)) \end{cases} \tag{3.18}
$$

where $\mathrm{d}_{s,t}(i,j)$ is the distance up to word i of s and word j of t, and $1_{\{s(i)\neq t(j)\}}$ the indicator function, being 1 if word i of s and word j of t do not match. $\mathit{leven}(s(i),t(j))$ denotes the Levenshtein distance between words $s(i)$ and $t(j)$.

The minimum distance d^* can be obtained efficiently via dynamic programming. We reconstruct the pairs of matching words, *i.e.* where $s(i) = t(j)$, by tracing the path of operations backwards through the cost matrix. Based on the matching words, a correspondence between subtitle and transcript lines can be established (*cf*. Fig. 3.3), and the identity transferred from the transcript to corresponding subtitle line. Subtitle lines for which the correspondence is ambiguous or which only consist of a single word are ignored.

Over the first 6 episodes of BBT, this method is able to assign a total of 2959 identities to subtitle lines, of which 2956 are correct, corresponding to a precision of 99.9% and a recall of 91.2%.

Speaking face detection The labeled speaker intervals (s_i, e_i, y_i) so far only indicate that a specific character y_i is currently speaking. However, it is very common that multiple

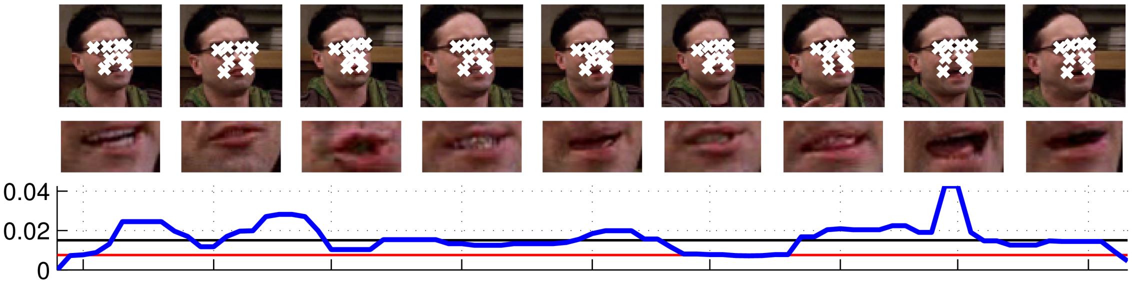


Figure 3.4.: Speaking-face detection: Is this face speaking or not? We first detect facial landmarks (*top*), crop out the mouth region (*middle*) and determine the normalized minimum distances of the mouth region patch to the respective previous frame (*bottom*, blue line). The distances are thresholded and accumulated to determine whether the face was speaking during the duration of a subtitle.

face tracks co-occur with the speaking interval. Therefore, we have to further determine which of the face tracks is indeed speaking. We detect the current speaker among the face tracks by analyzing the mouth movement.

To that end, we estimate the motion of the mouth region via block matching. The normalized distance between the motion-compensated blocks averaged over the speaker interval serves as confidence value which is thresholded to determine a face to be speaking or not-speaking (as a third possibility we have a region of confidence where we do not make a decision). Figure 3.4 shows a visualization of the speaking-face detection procedure.

The combination of the subtitle-transcript alignment and the visual speaker detection allows us to assign identities to some of the tracks. Despite the near-perfect results from the subtitle-transcript alignment, we do not obtain perfectly clean labels from this method, since the speaking face detection is noisy, Table 3.1 shows the precision and recall of the speaker assignment method on our data set. “#speaking tracks” denotes the number of tracks which were determined as speaking, which is usually less than 30% of the tracks (not all characters speak at the same time). On average, we associate an identity to about 16% of the tracks with a precision of 90% (BBT) and 86% (BUFFY), which is similar to the reported performances of Sivic et al. (2009). The lower speaker recall compared to Sivic et al. (2009) can be attributed to the higher number of total face tracks which we obtain from our more robust tracking method.

character	# face tracks	# assigned as speaking	speaker precision	speaker recall	#correct/wrong
Leonard	1146	218	90.83	17.28	
Sheldon	998	250	89.60	22.44	
Penny	525	97	92.78	17.14	
Unknown	415	10	90.00	2.17	
Howard	304	46	89.13	13.49	
Raj	291	27	92.59	8.59	
Mary	98	39	84.62	33.67	
Leslie	83	10	80.00	9.64	
Kurt	32	3	33.33	3.12	
Gablehauser	16	2	100.00	12.50	
Doug	8	0	–	–	
Summer	5	1	0.00	0.00	

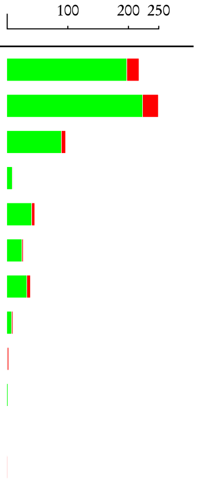


Table 3.2.: Character statistics and speaker assignment performance for BBT. For the 5 major characters, precision is around 90%. The absolute number of labeled tracks varies however, with *Raj* being the lowest with 27 tracks, since he speaks less than other characters.

Analyzing the performance on a character basis (see Tbl. 3.2 and Fig. B.1 for BBT, and Tbl. 3.3 and Fig. B.2 for BUFFY), we observe that both the absolute number of labeled tracks and the identity assignment performance varies between characters. For example, only 27 tracks are assigned the identity *Raj* (with a precision of 92.6%), whereas 218 tracks are assigned the identity *Leonard* (with a precision of 90.8%).

Face descriptors We employ a local-appearance-based method based on the discrete cosine transform (DCT) (Ekenel and Stiefelhagen, 2006) as face descriptor. Using the eye-center and mouth-center locations from the facial landmarks, the face is first aligned to a canonical pose via an affine transformation and cropped to a size of 48 × 64 pixels. If eye- or mouth-center cannot be determined, for example due to a profile view of the face, we crop a region around the track's bounding box in a best effort to obtain a good face patch. Otherwise, tracks which only show a profile view of the face might end up with no feature to identify them. The aligned face is then split into 6 × 8 blocks. For each block, the DCT is computed, of which we ignore the 0th value (average brightness) and retain the next five coefficients. Concatenating over all 48 blocks, we thus obtain a 240 dimensional feature vector for each frame in the track.

character	# face tracks	# assigned as speaking	speaker precision	speaker recall
Buffy	1324	288	90.28	19.64
Riley	586	77	88.31	11.60
Xander	555	122	77.05	16.94
Willow	488	108	90.74	20.08
Unknown	383	15	80.00	3.13
Giles	366	66	78.79	14.21
Dawn	360	44	86.36	10.56
Anya	299	53	81.13	14.38
Tara	257	33	78.79	10.12
Spike	239	50	90.00	18.83
Harmony	197	75	88.00	33.50
Xander2	154	15	93.33	9.09
Joyce	114	31	83.87	22.81
Glory	88	27	100.00	30.68
Dracula	70	5	80.00	5.71
Maclay	63	17	76.47	20.63
Beth	52	17	94.12	30.77
Graham	45	15	73.33	24.44
Overheiser	41	8	50.00	9.76
Mort	38	4	50.00	5.26
Leiach	33	0	–	–
Donny	32	8	87.50	21.88
Manager	27	6	66.67	14.81
Ben	23	7	85.71	26.09
Watchman	12	5	100.00	41.67
Sandy	9	2	100.00	22.22
Toth	6	0	–	–
Monk	0	0	–	–

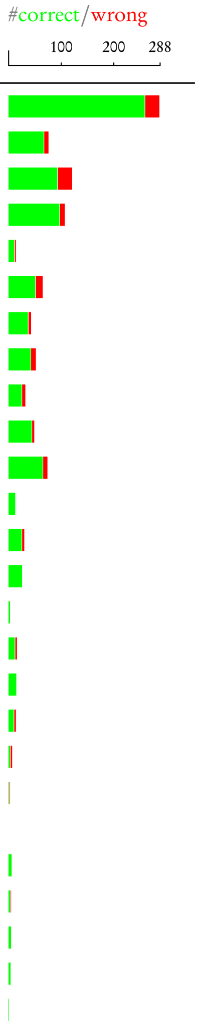


Table 3.3.: Character statistics and speaker assignment performance for BUFFY. For some background characters, we do not obtain any correct label, *e.g.* if they never speak or we cannot detect their respective tracks as speaking. The precision of the assignment varies between characters and has to be taken into account when judging later identification performance.

While this DCT-based facial descriptor is not among the state-of-the-art descriptors, it has proven robust in a series of previous work (Bäuml et al., 2010a; Bernardin et al., 2008; Ekenel et al., 2007c; Ekenel and Stiefelhagen, 2006; Fischer et al., 2010; Stallkamp et al., 2007) and is very efficient to compute.

We will perform all experiments with the same underlying features in order to ensure a fair comparison. Of course, a better alignment and more descriptive features are expected to have a positive influence on the recognition performance, and their incorporation into the proposed method should be the subject of future work.

Unlabeled data With only 16% of the face tracks labeled by the speaker assignment, we are left with 84% of the data that has no labels associated with it. This data constitutes our unlabeled feature set $\mathcal{X}_u$.

Constraints Constraints between training samples can be automatically deduced on the basis of face tracks. Positive constraints are formed, if two features stem from the same face track, based on the assumption that the tracker followed the face correctly. Negative constraints are formed when two tracks overlap temporally, based on the assumption that the same person cannot appear twice at the same time. These negative constraints are equivalent to the uniqueness or cannot-link constraint as used in other previous work (*e.g.*, Yan et al. (2006)).

Both negative and positive links can be constructed between all pairs of involved features. Thus, a track of length N induces $N \cdot (N-1)$ positive constraints. Two overlapping tracks with N and M features, respectively, induces $N \cdot M$ negative constraints between the respective features, even if they only overlap for one frame, due to the transitivity of positive relations within a track.

This automatic approach fails for negative links if the same person appears twice in a frame, *e.g.* due to a mirror, and for positive links when a track switch occurs. However, such events are rare compared to the errors made by the automatic speaker-based labeling: there is one scene involving a mirror in BUFFY, and the number of tracks switches is in the order of the number of *bad mismatches*, *i.e.* below 10 for every episode (*cf*. Tables 2.3 and 2.4).

3.3.3. Identification

Once we trained the classifier through minimization of Eq. 3.4, we can now identify every track. Again based on the assumption that all frames of a track stem from the same person, we perform a joint decision over all frames of a track.

In order to determine the identity y_t of a face track t with features $\{\mathbf{x}_i^{(t)}\}_{i=1}^{|t|}$, we score each frame according to Eq. 3.7. The individual scores are the averaged, leading to the following track score for identity k:

$$p_t(k) = \frac{1}{|t|}\sum_{i=1}^{|t|} P(y = k|\mathbf{x}_i^{(t)}) = \frac{1}{|t|}\sum_{i=1}^{|t|} \frac{e^{\theta_k^T \mathbf{x}_i^{(t)}}}{\sum_z e^{\theta_z^T \mathbf{x}_i^{(t)}}} . \tag{3.19}$$

The track is assigned the identity k with the overall highest score

$$s_t = \max_k p_t(k) \qquad \text{and} \qquad y_t = \operatorname*{argmax}_k p_t(k) \quad . \tag{3.20}$$

The outputs of the MLR classifier are in the range of 0 to 1 and could be interpreted as probabilities. We fuse the individual frame scores by taking the sum instead of the product over all frames, since we found this to be more robust to outliers in practice (a theoretical justification can be found in Kittler et al. (1998)).

Assignment to "unknown" Some characters which remain unnamed in the plot (*unknowns*) have small speaking roles. We can assign the "unknown" identity automatically to some of their face tracks with the described speaker-detection method and therefore we can automatically collect some training samples for them. In contrast to a "normal" character, we model all unknown characters as one *joint* class in the model. That is, features are collected from all unknowns and used as joint training data for the "unknown" class.

During identification, a track is assigned the "unknown" identity when it is the most likely class according to Eq. 3.20.

3.4. Evaluation

In the following, we evaluate our approach on the described data set.

Evaluation metrics A number of different evaluation criteria are established for the task of person identification.

As most comprehensive measure, we compute the track accuracy of the assignment, *i.e.* the number of correct assignments normalized by the number of total assignments:

$$\mathrm{ACC}_{\mathrm{track}} = \frac{1}{|T|} \sum_{t \in T} \mathbf{1}[y_t = \mathrm{gt}_t] \quad , \tag{3.21}$$

where y_t is the assigned identity according to Eq. 3.20 and gt_t the ground truth identity of track t. This measure does not take into account the track length. For example, one could argue that it is worse to make a wrong decision for a long track (whose identity will be displayed for a longer period of time in an end-user system) than for a short track. We therefore also compute the accuracy on a frame basis:

$$\begin{aligned} \mathrm{ACC}_{\mathrm{frame}} &= \frac{1}{\sum_{t \in T} |t|} \sum_{t \in T} \sum_{i=1}^{|t|} \mathbf{1}[y_t = \mathrm{gt}_t] \\ &= \frac{1}{\sum_{t \in T} |t|} \sum_{t \in T} |t| \cdot \mathbf{1}[y_t = \mathrm{gt}_t] \quad , \end{aligned} \tag{3.22}$$

where $|t|$ denotes the length of track t.

Everingham et al. (2006) and Sivic et al. (2009) report performance in terms of precision and recall. Their motivation is that by taking the classifier score as a confidence value, one can refuse to make a prediction when the confidence for the decision is too low. Let s be the score cutoff after which we refuse to make a prediction, and $T(s) = \{t \in T, \mathrm{score}(t) > s\}$, then

$$PR(s) = \frac{1}{|T(s)|} \sum_{t \in T(s)} \mathbf{1}[y_t = \mathrm{gt}_t] \tag{3.23}$$

$$REC(s) = \frac{1}{|T|} \sum_{t \in T(s)} \mathbf{1}[y_t = \mathrm{gt}_t] \tag{3.24}$$

The *average precision* (AP) is then defined as the area under the precision recall curve.

Due to the *unknown* characters, the identification is an *open set* recognition problem. We therefore also report

	1	2	3	4	5	6	Avg.
BBT							
$\mathscr{L}_l$	89.04	88.46	73.94	77.16	78.63	62.98	78.37
$\mathscr{L}_l + \mathscr{L}_u$	89.04	88.46	73.79	77.49	78.63	63.10	78.42
$\mathscr{L}_l + \mathscr{L}_c$	**89.65**	**90.24**	**76.36**	78.14	**81.49**	65.69	80.26
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	89.50	**90.24**	**76.36**	**78.79**	81.30	**65.92**	**80.35**
BUFFY							
$\mathscr{L}_l$	77.39	69.12	74.62	75.33	71.43	68.32	72.70
$\mathscr{L}_l + \mathscr{L}_u$	77.76	68.92	74.62	75.22	71.31	68.41	72.71
$\mathscr{L}_l + \mathscr{L}_c$	**78.64**	**71.22**	**75.71**	75.67	73.45	**69.74**	**74.07**
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	78.52	71.12	75.46	**75.78**	**73.69**	69.65	74.04

Table 3.4.: Face recognition results using MLR and our loss extensions for BBT. While the entropy loss only provides a marginal improvement, the constraints improve performance by about 2% for BBT and 1.3% for BUFFY.

- *Correct Classification Rate* (CCR), the identification performance among the known characters:

$$\mathrm{CCR} = \frac{1}{|T_{(\mathrm{known})}|} \sum_{t \in T_{(\mathrm{known})}} \mathbf{1}[y_t = \mathrm{gt}_t] \quad , \tag{3.25}$$

- *False Acceptance Rate* (FAR): A false acceptance is the incorrect classification of an unknown as one of the known characters:

$$\mathrm{FAR} = \frac{1}{|T_{(\mathrm{unknown})}|} \sum_{t \in T_{(\mathrm{unknown})}} \mathbf{1}[y_t \neq \mathbf{u}] \quad . \tag{3.26}$$

- *False Rejection Rate* (FRR): A false rejection is the incorrect classification of one of the known characters as unknown:

$$\mathrm{FAR} = \frac{1}{|T_{(\mathrm{known})}|} \sum_{t \in T_{(\mathrm{known})}} \mathbf{1}[y_t = \mathbf{u}] \quad . \tag{3.27}$$

3.4.1. Results and Analysis

In all experiments we employ a polynomial kernel of degree 2, *i.e.* $k(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^2$. In order to fit the kernel's Gram matrix into memory, we subsample up to 5000 features

	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6	BF Avg.
SUM (Sivic et al., 2009)	0.89	0.83	0.68	0.82	0.85	0.69	0.79
MKL (Sivic et al., 2009)	0.90	0.83	0.70	0.86	0.85	0.70	0.81
VF^2 (Parkhi et al., 2014)	0.94	0.83	0.78	0.89	0.92	0.74	0.85
MLR	0.93	0.81	0.83	0.87	0.91	0.86	0.87
MLR*	0.96	0.84	0.92	0.93	0.96	0.92	0.92

Table 3.5.: Comparison with the approaches by Sivic et al. (2009) and Parkhi et al. (2014) in terms of average precision as reported in the respective papers. The comparison is performed on Sivic et al. (2009)'s tracks. Training and testing is performed on each episode separately following the protocol of the original papers. *MLR** denotes performance when taking into account all episodes jointly for training. For joint training, Sivic et al. (2009) report an increase in AP to 0.82 and 0.79 for BF-3 and BF-6, respectively.

per character and further restrict the kernel basis to a maximum of 20000 features (Lee and Mangasarian, 2001) (*cf*. Sec. 5.2.1). If not otherwise specified, we employ 50000 unlabeled features and 100000 constraint pairs. We select parameters λ, μ and γ on a separate validation set, consisting of episodes 12-15 of each series.

In the following, we analyze the performance of our approach.

Comparison of influence of different loss terms We compare the recognition performance by incorporating the different loss terms in Tbl. 3.4 (for details see Tables 3.7 (BBT) and 3.7 (BUFFY)). The incorporation of the entropy loss $\mathscr{L}_u$ for unlabeled data only provides a marginal improvement. A possible explanation is that the data obtained from the automatic speaker assignment is already well-defined enough to place decision boundaries at correct locations. We selected unlabeled features by subsampling (see above).

A possible avenue for future work might be to select unlabeled features with a more sophisticated strategy to improve their impact. Constraints on the other hand have a more visible influence. The models learned by including the constraint loss $\mathscr{L}_c$ show a 2% (respectively 1.3%) improvement in track-level accuracy over when using only the supervised loss. A similar increase in performance can also be observed in frame-level accuracy and correct classification rate. For BBT, the inclusion of the constraint loss is also able to reduce the false acceptance rate, whereas this is not the case for BUFFY.

Episode	1	2	3	4	5	6	Avg.
Speaker				BBT			
automatic	89.50	90.24	76.36	78.79	81.30	**65.92**	80.35
ground truth	**94.52**	**93.50**	**77.42**	**83.52**	**82.25**	62.40	**82.27**
				BUFFY			
automatic	78.52	71.12	75.46	75.78	73.69	69.65	74.04
ground truth	**82.91**	**80.28**	**82.50**	**79.33**	**77.98**	**75.33**	**79.72**

Table 3.6.: Influence of the noisy speaker assignment. Correcting the speaker assignments to the ground truth track identities improves performance for both BBT and BUFFY. The relative improvement for BUFFY is higher, possibly since the original speaker labels were noisier.

Comparison with related work We compare our approach with baseline approaches from (Sivic et al., 2009) and (Parkhi et al., 2014) and report results in Table 3.5. As (Sivic et al., 2009) is an extension and improvement of (Everingham et al., 2006), we do not explicitly compare with the latter. We evaluate on their tracks (*cf*., Table 2.8) to perform a fair comparison. The rest of our pipeline is as described above. Following their evaluation protocol, we train and test on each episode separately and do not count speaker-assigned tracks in the evaluation. We report results in terms of average precision to be comparable with the reported results from (Sivic et al., 2009) and (Parkhi et al., 2014), respectively.

Our approach (denoted *MLR* in Table 3.5) outperforms both other approaches with a mean average precision of 0.87. By training and testing on each episode separately, some characters are underrepresented in terms of available training data for some episodes. If we take training data from all episodes into account and train a joint model over all episodes, mean average precision increases to 0.92 (method denoted as *MLR**). Using training data from all episodes, Sivic et al. (2009) report for their approach an increase in AP to 0.82 and 0.79 for BF-3 and BF-6, respectively.

Correcting speaker labels The speaker assignment (Sec. 3.3.2) is not perfect and the label accuracy is only around 80-90% (*cf*., Tbl. 3.1). In order to analyze the influence of these errors, we correct all speaker assignments to the underlying true track identity. As can be seen in Tbl. 3.6, the correction of the speaker labels has a positive influence on the recognition performance. The impact on BUFFY is higher, possibly since the original labels were noisier. However, this experiment also shows, that only increasing precision is not enough. A more accurate speaker assignment method could be beneficial, but ideally it would not only increase precision of the labels but also the recall, labeling more of the possible speaker tracks.

Track length In Fig. 3.5 (a,b) we analyze the recognition performance in dependency of the track length. We show both the absolute number of correctly/wrongly labeled tracks (Fig. 3.5 (a)) as well as the relative ranks of the correct track depending on the track length (Fig. 3.5 (b)). Similar to what was one of our motivations for a better tracker, we observe that with increasing track length the recognition rate rises consistently. For tracks shorter than 25 frames, we obtain recognition accuracies below 60%, while for tracks with a length above 75 frames, the recognition accuracy is around 80%.

However, we also observe that a large number of tracks is actually shorter than 75 frames. In part, this is due to the non-perfect tracker which still breaks tracks within a shot due to difficult poses or short-term occlusions. On the other hand, the track length is also bounded by the shot length, and cannot be extended indefinitely even by a perfect tracker.

Face size In Fig. 3.5 (c,d) we analyze the dependency of track-level accuracy on the mean face size of a track. As expected, a small face is generally harder to recognize than a larger face, with accuracies around 45% for faces of size below 25px up to over 90% starting with faces above 100px in width.

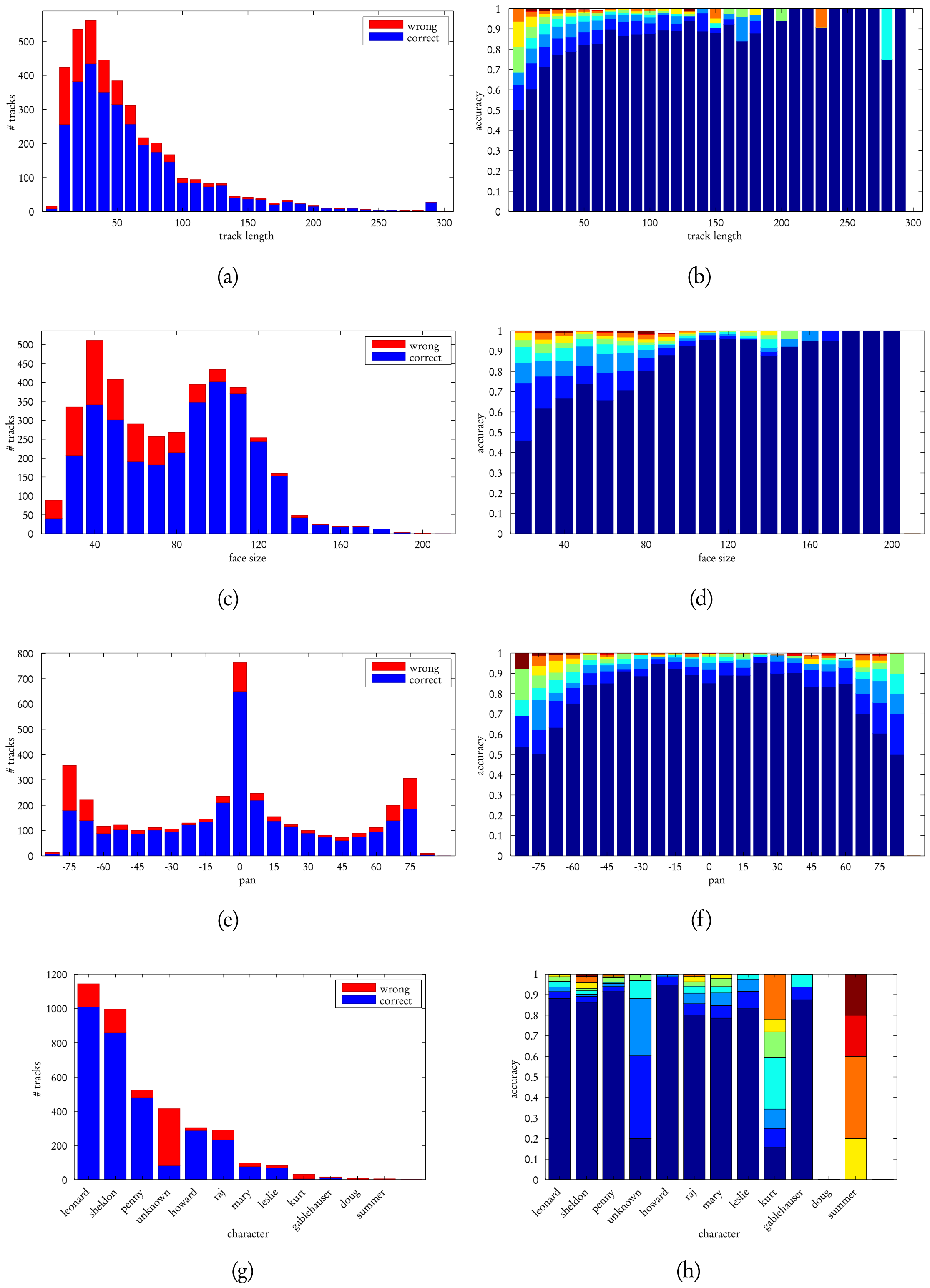
Interesting to note is also the distribution of the absolute number of tracks over face size (Fig. 3.5 (c)). There are two peaks around 40px and 100px. This is due to the filming style in BBT with basically two camera settings, one for wide angle overview shots, and one for close-up conversation shots. While we resize every face to a canonical size of 48x64px after alignment, it might be an interesting avenue for future work to exploit such filming styles and for example learn two different models, one for higher and one for lower resolution faces.

Face pose We further analyze the dependency of the track-level accuracy on the head pose, namely the mean pan angle of the track. In Fig. 3.5 (b), we see that accuracy consistently decreases with higher pan angles. This is not surprising, since our affine face alignment is not able to remove the changes in appearance induce by out-of-plane face rotation. We also observe a small reduction in accuracy around the frontal pose. One possible reason for this might be that many small faces, *e.g.* from extras in the background of wide angle shots, are mostly frontal faces. However, for both small faces as well as unknowns, recognition accuracy is lower than average, and influence the frontal performance as well.

Characters The recognition accuracies across different characters can be found in Fig. 3.5 (g,h). Unknown characters, although they present the fourth largest number of tracks, are only very poorly recognized. The true identification rate for unknowns lies below 10%. This is also reflected in a high false acceptance rate. One possible reason is that they are underrepresented in the training set with only 10 tracks labeled as unknown over the 6 episodes (*cf*., Tbl. 3.2). Main characters are recognized well with around 90% accuracy each. If the number of training samples is low for a character, the model deteriorates and so does the model's performance, for example for Kurt, Gablehauser, Doug and Summer. Also for Raj, despite being a main character, fewer training samples are available since he does not speak very often (*cf*., Table 3.2).

Character confusion matrix Confusion matrices of the character assignments can be found in Figures 3.7 and 3.8 for BBT and BUFFY, respectively. As in Fig. 3.5 (g,h), we observe that *unknowns* are confused often with other known characters. Also, for characters for which the speaker assignment performed poorly, exhibit more confusion with other characters. This was expected and motivates the investigation of better visual speaker detection methods in future work.

Dependency on the number of constraints See Fig. 3.6 for an analysis of the influence of the number of considered constraints on the accuracy. In this setting, we do not consider any unlabeled samples. We train model on BBT and BUFFY for using different numbers of constraints, ranging from as few as 1000 constraints to 10000000 constraints. Both frame- and track-level accuracy benefit from a higher number of constraints.



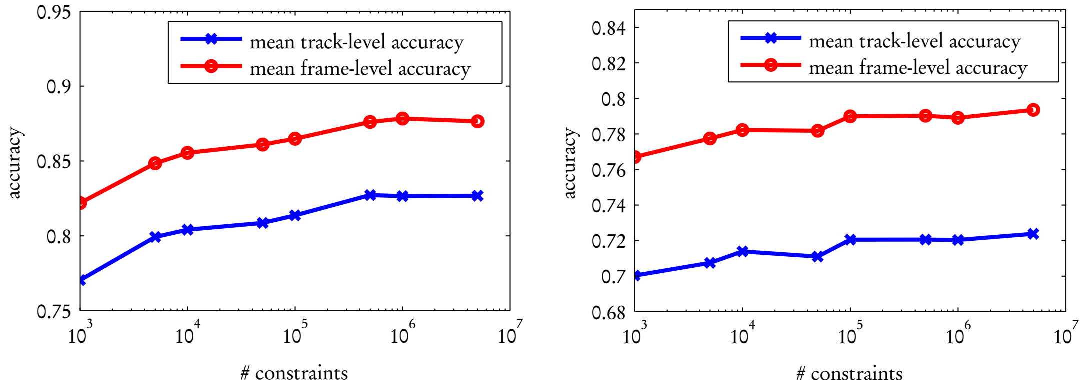
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3.5.: Track-level recognition accuracy on BBT in dependency of (a,b) track length, (c,d) face size, (e,f) face pose and (g,h) the character. The left column shows in each case the absolute numbers of correctly/wrongly labeled tracks, while the right column shows the relative performance over multiple ranks.



(a) BBT

(b) BUFFY

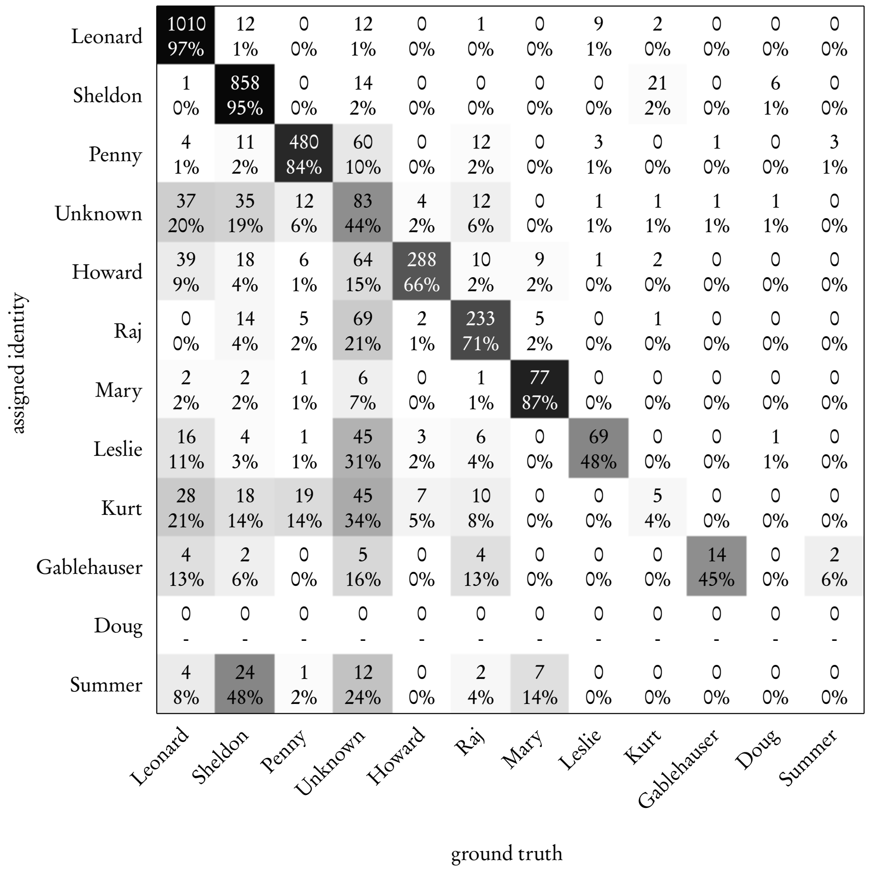
Figure 3.6.: Recognition accuracy in dependency of the number of constraints employed during training for (a) BBT and (b) BUFFY. Both track-level and frame-level accuracy improve with a higher number of constraints.

	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	Avg.
Track-level Accuracy							
$\mathscr{L}_l$	89.04	88.46	73.94	77.16	78.63	62.98	78.37
$\mathscr{L}_l + \mathscr{L}_u$	89.04	88.46	73.79	77.49	78.63	63.10	78.42
$\mathscr{L}_l + \mathscr{L}_c$	**89.65**	**90.24**	**76.36**	78.14	**81.49**	65.69	80.26
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	89.50	**90.24**	**76.36**	**78.79**	81.30	**65.92**	**80.35**
Frame-level Accuracy							
$\mathscr{L}_l$	91.34	91.84	76.47	85.27	80.70	73.32	83.16
$\mathscr{L}_l + \mathscr{L}_u$	91.34	91.84	76.43	85.46	80.70	73.18	83.16
$\mathscr{L}_l + \mathscr{L}_c$	**94.49**	**94.70**	80.77	87.02	**85.09**	76.98	86.51
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	94.35	**94.70**	**80.80**	**87.35**	84.93	**77.27**	**86.57**
Correct Classification Rate							
$\mathscr{L}_l$	92.13	92.03	88.91	88.33	88.04	82.98	88.74
$\mathscr{L}_l + \mathscr{L}_u$	92.13	92.03	88.91	88.53	88.04	82.82	88.75
$\mathscr{L}_l + \mathscr{L}_c$	**95.05**	**94.89**	**91.35**	90.35	**91.57**	85.76	91.49
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	94.91	**94.89**	91.31	**90.69**	91.39	**85.99**	**91.53**
False Acceptance Rate							
$\mathscr{L}_l$	56.87	**100.00**	98.34	**94.10**	92.87	83.37	87.59
$\mathscr{L}_l + \mathscr{L}_u$	56.87	**100.00**	98.62	**94.10**	92.87	83.37	87.64
$\mathscr{L}_l + \mathscr{L}_c$	**39.62**	**100.00**	82.82	98.97	**79.70**	74.50	79.27
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	**39.62**	**100.00**	**82.43**	98.97	**79.70**	**73.88**	**79.10**
False Rejection Rate							
$\mathscr{L}_l$	**0.00**	**0.56**	**0.00**	0.19	**0.00**	0.90	0.27
$\mathscr{L}_l + \mathscr{L}_u$	**0.00**	**0.56**	**0.00**	**0.07**	**0.00**	0.90	**0.26**
$\mathscr{L}_l + \mathscr{L}_c$	1.40	2.72	2.98	0.08	2.26	**0.63**	1.68
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	1.54	2.60	2.99	0.08	2.26	0.64	1.68

Table 3.7.: Result details for BBT.

	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6	Avg.
			Track-level Accuracy				
$\mathscr{L}_l$	77.39	69.12	74.62	75.33	71.43	68.32	72.70
$\mathscr{L}_l + \mathscr{L}_u$	77.76	68.92	74.62	75.22	71.31	68.41	72.71
$\mathscr{L}_l + \mathscr{L}_c$	**78.64**	**71.22**	**75.71**	75.67	73.45	**69.74**	**74.07**
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	78.52	71.12	75.46	**75.78**	**73.69**	69.65	74.04
			Frame-level Accuracy				
$\mathscr{L}_l$	82.20	77.45	77.91	83.03	80.15	75.06	79.30
$\mathscr{L}_l + \mathscr{L}_u$	82.67	77.50	77.77	82.96	80.02	**75.12**	79.34
$\mathscr{L}_l + \mathscr{L}_c$	83.87	**78.66**	80.53	83.19	82.55	74.78	80.60
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	**84.18**	78.66	**80.57**	**83.26**	**82.85**	74.53	**80.67**
			Correct Classification Rate				
$\mathscr{L}_l$	82.35	82.68	77.89	85.55	85.17	78.09	81.95
$\mathscr{L}_l + \mathscr{L}_u$	82.83	82.81	77.75	85.48	85.17	**78.15**	82.03
$\mathscr{L}_l + \mathscr{L}_c$	84.04	**84.68**	80.53	85.71	87.17	77.79	83.32
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	**84.35**	84.67	**80.57**	**85.79**	**87.46**	77.53	**83.39**
			False Acceptance Rate				
$\mathscr{L}_l$	**59.69**	**71.09**	**19.87**	**100.00**	72.47	**94.32**	69.57
$\mathscr{L}_l + \mathscr{L}_u$	**59.69**	71.74	**19.87**	**100.00**	74.04	**94.32**	69.94
$\mathscr{L}_l + \mathscr{L}_c$	**59.69**	77.09	**19.87**	**100.00**	65.94	**94.32**	69.49
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	**59.69**	77.09	**19.87**	**100.00**	**65.49**	**94.32**	**69.41**
			False Rejection Rate				
$\mathscr{L}_l$	0.90	**0.66**	**0.02**	**0.10**	**0.00**	**0.63**	**0.39**
$\mathscr{L}_l + \mathscr{L}_u$	0.90	**0.66**	**0.02**	**0.10**	**0.00**	**0.63**	**0.39**
$\mathscr{L}_l + \mathscr{L}_c$	0.91	0.94	0.51	0.46	0.03	0.81	0.61
$\mathscr{L}_l + \mathscr{L}_u + \mathscr{L}_c$	**0.80**	0.95	0.53	0.46	0.03	0.81	0.60

Table 3.8.: Result details for BUFFY.



assigned identity
ground truth

	Leonard	Sheldon	Penny	Unknown	Howard	Raj	Mary	Leslie	Kurt	Gablehauser	Doug	Summer
Leonard	1010 97%	12 1%	0 0%	12 1%	0 0%	1 0%	0 0%	9 1%	2 0%	0 0%	0 0%	0 0%
Sheldon	1 0%	858 95%	0 0%	14 2%	0 0%	0 0%	0 0%	0 0%	21 2%	0 0%	6 1%	0 0%
Penny	4 1%	11 2%	480 84%	60 10%	0 0%	12 2%	0 0%	3 1%	0 0%	1 0%	0 0%	3 1%
Unknown	37 20%	35 19%	12 6%	83 44%	4 2%	12 6%	0 0%	1 1%	1 1%	1 1%	1 1%	0 0%
Howard	39 9%	18 4%	6 1%	64 15%	288 66%	10 2%	9 2%	1 0%	2 0%	0 0%	0 0%	0 0%
Raj	0 0%	14 4%	5 2%	69 21%	2 1%	233 71%	5 2%	0 0%	1 0%	0 0%	0 0%	0 0%
Mary	2 2%	2 2%	1 1%	6 7%	0 0%	1 1%	77 87%	0 0%	0 0%	0 0%	0 0%	0 0%
Leslie	16 11%	4 3%	1 1%	45 31%	3 2%	6 4%	0 0%	69 48%	0 0%	0 0%	1 1%	0 0%
Kurt	28 21%	18 14%	19 14%	45 34%	7 5%	10 8%	0 0%	0 0%	5 4%	0 0%	0 0%	0 0%
Gablehauser	4 13%	2 6%	0 0%	5 16%	0 0%	4 13%	0 0%	0 0%	0 0%	14 45%	0 0%	2 6%
Doug	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -
Summer	4 8%	24 48%	1 2%	12 24%	0 0%	2 4%	7 14%	0 0%	0 0%	0 0%	0 0%	0 0%

Figure 3.7.: Confusion matrix for BBT results. As can be seen also in Fig. 3.5 (g), *unknowns* are confused most often with other known characters. Despite having 97% precision own his assigned tracks, *Leonard* collects quite a few tracks from other characters. Comparing these results with the speaker assignment performance (Tbl. 3.2), we observe as expected that those characters which were assigned only a few labels or whose assignments had low precision, also perform worse in the actual identification.

assigned identity

ground truth

	Buffy	Riley	Xander	Willow	Unknown	Giles	Dawn	Anya	Tara	Spike	Harmony	Xander2	Joyce	Glory	Dracula	Maclay	Beth	Graham	Overheiser	Mort	Leiach	Donny	Manager	Ben	Watchman	Sandy	Toth
Buffy	1059 96%	3 0%	3 0%	3 0%	8 1%	2 0%	5 0%	3 0%	1 0%	1 0%	4 0%	1 0%	4 0%	0 0%	1 0%	1 0%	1 0%	0 0%	0 0%	0 0%	2 0%	0 0%	1 0%	0 0%	0 0%	0 0%	1 0%
Riley	33 5%	469 78%	5 1%	3 0%	29 5%	11 2%	1 0%	5 1%	13 2%	1 0%	1 0%	2 0%	9 1%	1 0%	1 0%	4 1%	1 0%	4 1%	1 0%	0 0%	3 0%	3 0%	1 0%	0 0%	2 0%	2 0%	0 0%
Xander	37 5%	18 3%	480 67%	11 2%	47 7%	1 0%	9 1%	3 0%	11 2%	2 0%	2 0%	72 10%	5 1%	2 0%	8 1%	1 0%	0 0%	0 0%	3 0%	1 0%	1 0%	2 0%	1 0%	1 0%	0 0%	0 0%	0 0%
Willow	6 1%	9 2%	6 1%	400 85%	18 4%	2 0%	5 1%	4 1%	1 0%	0 0%	2 0%	4 1%	2 0%	1 0%	2 0%	3 1%	0 0%	0 0%	1 0%	1 0%	0 0%	0 0%	1 0%	0 0%	0 0%	0 0%	0 0%
Unknown	3 3%	0 0%	7 7%	0 0%	62 62%	1 1%	6 6%	2 2%	3 3%	1 1%	2 2%	1 1%	0 0%	0 0%	4 4%	0 0%	0 0%	1 1%	0 0%	0 0%	6 6%	0 0%	0 0%	1 1%	0 0%	0 0%	0 0%
Giles	21 5%	24 5%	6 1%	10 2%	27 6%	320 69%	13 3%	5 1%	7 2%	4 1%	0 0%	5 1%	3 1%	2 0%	1 0%	3 1%	0 0%	4 1%	1 0%	1 0%	2 0%	1 0%	0 0%	2 0%	0 0%	0 0%	0 0%
Dawn	27 7%	12 3%	5 1%	5 1%	25 7%	2 1%	275 75%	2 1%	6 2%	0 0%	2 1%	2 1%	1 0%	2 1%	0 0%	0 0%	0 0%	0 0%	1 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Anya	45 12%	4 1%	8 2%	10 3%	16 4%	3 1%	13 3%	244 65%	3 1%	0 0%	4 1%	3 1%	6 2%	6 2%	1 0%	2 1%	0 0%	0 0%	3 1%	2 1%	1 0%	1 0%	3 1%	0 0%	0 0%	0 0%	0 0%
Tara	16 7%	6 2%	1 0%	4 2%	11 4%	1 0%	6 2%	5 2%	181 74%	0 0%	4 2%	0 0%	3 1%	0 0%	2 1%	1 0%	2 1%	0 0%	0 0%	1 0%	1 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Spike	22 5%	12 3%	13 3%	8 2%	64 15%	11 3%	7 2%	5 1%	6 1%	217 52%	8 2%	8 2%	2 0%	6 1%	4 1%	0 0%	0 0%	0 0%	7 2%	0 0%	7 2%	1 0%	0 0%	4 1%	2 0%	0 0%	5 1%
Harmony	7 3%	3 1%	2 1%	3 1%	16 7%	2 1%	9 4%	1 0%	11 5%	5 2%	156 70%	0 0%	2 1%	1 0%	0 0%	0 0%	0 0%	0 0%	2 1%	2 1%	1 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Xander2	0 0%	0 0%	1 2%	0 0%	0 0%	0 0%	0 0%	2 4%	0 0%	0 0%	0 0%	47 94%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Joyce	9 7%	4 3%	3 2%	7 5%	6 5%	0 0%	5 4%	9 7%	2 2%	0 0%	7 5%	0 0%	72 54%	0 0%	1 1%	0 0%	0 0%	0 0%	0 0%	3 2%	0 0%	0 0%	3 2%	1 1%	1 1%	0 0%	0 0%
Glory	1 1%	0 0%	1 1%	4 4%	5 5%	1 1%	0 0%	5 5%	6 7%	0 0%	0 0%	0 0%	2 2%	66 72%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	1 1%	0 0%
Dracula	12 18%	3 4%	0 0%	5 7%	0 0%	1 1%	0 0%	1 1%	1 1%	0 0%	0 0%	0 0%	0 0%	0 0%	45 66%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Maclay	0 0%	2 4%	0 0%	2 4%	3 5%	2 4%	1 2%	0 0%	1 2%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	40 71%	2 4%	0 0%	0 0%	0 0%	2 4%	1 2%	0 0%	0 0%	0 0%	0 0%	0 0%
Beth	1 2%	0 0%	0 0%	0 0%	3 5%	1 2%	0 0%	0 0%	2 3%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	46 79%	0 0%	0 0%	0 0%	2 3%	0 0%	0 0%	3 5%	0 0%	0 0%	0 0%
Graham	4 6%	11 16%	1 1%	2 3%	10 15%	1 1%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	36 54%	1 1%	0 0%	0 0%	0 0%	0 0%	1 1%	0 0%	0 0%	0 0%
Overheiser	6 17%	1 3%	0 0%	1 3%	2 6%	0 0%	0 0%	0 0%	0 0%	2 6%	2 6%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	21 60%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Mort	4 8%	0 0%	0 0%	2 4%	6 12%	0 0%	3 6%	1 2%	0 0%	1 2%	3 6%	0 0%	1 2%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	27 56%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Leiach	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -
Donny	1 3%	0 0%	0 0%	1 3%	1 3%	2 5%	0 0%	0 0%	1 3%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	5 14%	0 0%	0 0%	0 0%	0 0%	3 8%	23 62%	0 0%	0 0%	0 0%	0 0%	0 0%
Manager	1 2%	3 7%	10 24%	0 0%	0 0%	0 0%	0 0%	1 2%	0 0%	0 0%	0 0%	9 22%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	17 41%	0 0%	0 0%	0 0%	0 0%
Ben	7 12%	2 3%	2 3%	6 10%	18 31%	1 2%	2 3%	0 0%	1 2%	5 8%	0 0%	0 0%	2 3%	0 0%	0 0%	3 5%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	10 17%	0 0%	0 0%	0 0%
Watchman	1 8%	0 0%	0 0%	1 8%	3 23%	0 0%	0 0%	1 8%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	7 54%	0 0%	0 0%
Sandy	1 7%	0 0%	1 7%	0 0%	3 20%	1 7%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	1 7%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	2 13%	0 0%	0 0%	0 0%	0 0%	6 40%	0 0%
Toth	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -

Figure 3.8.: Confusion matrix for BUFFY result. As for BBT, the bad performance of *unknown* recognition is quite visible in the confusion matrix. Similarly, those with poor speaker assignment performance show high confusion rates with other characters.

Chapter 4

Multimodal Person Identification

The identification of faces is an important step towards full person identification. However, faces alone are generally not sufficient to recognize all persons in a movie of TV series. On the one hand, we cannot assume that the face of the person is always localized by the tracker. A face might be (partially) occluded, in an uncommon pose or not visible at all, *e.g.* when the person is shown from behind. On the other hand, even if the face is localized, it might not contain enough information to distinguish between different persons. Fortunately, there are more cues in an image or video to identify a person than just the face, for example the hair, clothing, gait, gender, speech or even the estimated height and body size of a person. Humans, too, make use of such additional information for a better identification (Gallagher and Chen, 2008a; Kumar and Berg, 2009).

In this chapter, we extend the problem from recognizing *faces* to recognizing *persons*, even when the face is not visible. We perform person tracking in addition to face tracking to localize all person instances in the video. From the video, we can automatically obtain additional information such as clothing appearance, subtitles, audio or constraints between tracks. The arguably most important modality after faces is the clothing appearance of a person, since it is generally available, *i.e.*, it can be extracted for every person, compared for example to speech, which is only available for the current speaker. Given a face or person detection, a rough estimate of the clothing appearance is also relatively simple to extract. Further, clothing can be discriminative enough to differentiate between multiple persons[1], compared for example to gender, which constitutes only binary information.

[1] Of course, this is data dependent. For example, discriminating between the four *Teletubbies* via their color is simple, whereas discriminating between different instances of Mr. Smith in *The Matrix*, who all wear similar black suits with white shirts, is not.

In order to perform fully automatic recognition, we do not assume manual labels for person tracks to learn clothing models. As with faces, we are interested in automatically obtaining labels for learning clothing models. Since many of the person tracks can be associated with faces, we use the estimated face identities from the previous chapter to assign preliminary labels to clothing features. Of course, face identities are not 100% accurate and we therefore consider a soft labeling with different confidences for the given identities. The learned clothing models are then used to propagate clothing identities to other person tracks which do not have an associated face.

We are further interested in combining more information sources than just faces and clothing to perform the best identification. In contrast to the previous chapter, we will perform a late fusion of the different modalities. For the fusion step, each modality is regarded as a multi-class classifier which outputs confidences for each of the possible identities. Some modalities can be directly associated with a face or person, whereas others cannot (*e.g.*, subtitles). We therefore integrate them in different ways. Finally, the fusion is jointly performed over all tracks in a shot and takes into account additional information such as constraints between tracks.

4.1. Background and related work

Using other information than faces for person identification has been a research topic for a long time. In domains where a face is usually not visible with high enough resolution for identification such as camera networks, approaches have resorted to identification based on general appearance (*e.g.*, Farenzena et al. (2010); Gheissari et al. (2006); Gray and Tao (2008); Nakajima et al. (2003)) or gait (*e.g.* (Boyd and Little, 2005)). In the surveillance domain, the problem is often posed as a *re*-identification problem, *i.e.* given one or multiple instances of a person, the task is to find all other past and future appearances of the same person. For a recent overview over appearance-based person re-identification approaches see (Vezzani et al., 2013).

Clothing and appearance description Many different ways have been developed to describe the general appearance of a person.

A multitude of different descriptors has been tried and evaluated in different contexts. For example, simple histograms in different color and texture spaces have been shown to work quite reasonable, *e.g.* color histograms in YCbCr (Everingham et al., 2006), RGB (Anguelov et al., 2007), LSH (Shi et al., 2013), LCC (Gallagher and Chen, 2008a) or HSV (Jaffré and Joly, 2004) color spaces, and texture descriptors based on Gabor filter responses (Anguelov et al., 2007), LBP (Shi et al., 2013), Gaussian-based filter banks (Zhang et al., 2010), horizontal and vertical edge detector responses (Gallagher and Chen, 2008a), covariance matrices (Bak et al., 2011) or matches of local descriptors such as SIFT (Shi et al., 2013) or GLOH (Bäuml and Stiefelhagen, 2011).

The descriptor is often computed on a rectangular region below the face (*e.g.*, Anguelov et al. (2007); Everingham et al. (2006)) to avoid the need for a more sophisticated segmentation of the clothing region. But clothing and general appearance of a person contain more structure than can be captured with one global histogram. A more fine-grained segmentation of the torso of the person has been shown to improve recognition performance (Gallagher and Chen, 2008a; Sivic et al., 2006a)). Descriptors can for example also be extracted from upper and lower torso regions (Annesley et al., 2005; Weber et al., 2011), hair, face and upper body regions (Sivic et al., 2006a) or accumulated over local regions (Farenzena et al., 2010). By performing a full body-pose estimation first, the description can be attributed to individual body parts (Cheng et al., 2011). Body part descriptors can also be used to find more instances of the same person in other images (Sivic et al., 2006a) or adjacent frames (Ramanan et al., 2007a).

At a higher level, clothing can also be described in terms of the type of garment and its configuration. Aiming towards a more semantic description of clothing in surveillance footage, Borràs et al. (2003) match one of 5 graph structures against sub-segmentations of the clothing region and label nodes of the graph for example as tie or jacket. For a more fine-grained description of clothing in fashion photographs, Yamaguchi et al. (2012) assign one of 17 clothing and accessory labels such as dress, shirt, shoes or bag to a superpixels in an over-segmentation of the body. Similarly, high level attributes describing the style and type of clothing and soft-biometric cues such as gender and hair color, can be used in person re-identification tasks (Layne et al., 2012; Vaquero et al., 2009).

Instead of defining regions and features manually, different approaches learn appearance-based distance functions from training data. For the task of deciding for full person images from different viewpoints whether they show the same person or not, Gray and Tao (2008) define a feature pool from which features and feature regions are selected by boosting to learn a discriminative decision function. Similar approaches explored support vector ranking (Prosser et al., 2010) or metric learning (Dikmen et al., 2010;

Hirzer et al., 2012; Köstinger et al., 2012). Following recent advances in deep learning, the underlying features themselves can also be learned from data instead of defining a feature pool by hand (Li et al., 2014).

Fusion of different modalities Many approaches use other modalities than just faces alone. As another biometric cue, speech is often combined with visual cues (*e.g.*, Bernardin et al. (2008); Bredin et al. (2012); Ekenel et al. (2007a)). In the context of multimedia data and photo collections, clothing is another frequently used cue, albeit not uniquely tied to an identity (Anguelov et al., 2007; Everingham et al., 2006; Gallagher and Chen, 2008a; Ramanan et al., 2007a). Despite only being a binary cue, the gender of a person can be estimated and improve the identity decision (*e.g.*, Cour et al. (2009); Gallagher and Chen (2008b)). Even group relations in pictures (Gallagher and Chen, 2009) or the first name combined with the age of a person (Gallagher and Chen, 2008b) provide valuable information about a person.

In the simplest case, different modalities such as face and clothing are combined on feature level (*e.g.*, simply concatenating the individual descriptors) or on score level (*e.g.*, combining the scores of individual modality classifiers by a weighted sum) (Everingham et al., 2006; Kittler et al., 1998; Ramanan et al., 2007a).

An interesting approach for fusion of different modalities is presented by Markov random fields (MRFs). MFRs were first introduced by Geman and Geman (1984) and are used for diverse tasks such as segmentation (Boykov and Jolly, 2001; Gallagher and Chen, 2008a; Rother et al., 2004), image restoration/de-noising Geman and Geman (1984); Greig et al. (1989), depth-estimation (Boykov et al., 1998). An MRF describes the joint distribution of a set of random variables as an undirected graph, where dependencies between random variables are given by edges within the graph. By modeling each face as a random variable, a joint distribution over all face identities can be expressed, including interdependencies between faces.

As such, MRFs have been used for inference of identities in different approaches for person identification in personal photo collections (Anguelov et al., 2007; Gallagher and Chen, 2007; Lin et al., 2010). Gallagher and Chen (2007) focus on faces only. They incorporate a pairwise potential between nodes to model the similarity between faces in different images and to enforce uniqueness of identities in the same image. Anguelov et al. (2007) further include a pairwise potential to incorporate clothing similarity. In order to deal with changes in clothing, the timestamps of the photos are considered to learn clothing models for limited time intervals. Lin et al. (2010) extend their MRF model to recognize events and locations of the photos jointly with recognizing people. By establishing a person-location relationship, the presence of a person at a particular location can be inferred, and in turn be used for constraining the identification based on the present people.

4.1.1. Discussion and contribution

The inclusion of additional context in the identification decision has been validated by many approaches. Biometric cues such as speech clearly provide additional information on the unique identity of a person. However, also non-unique modalities such as clothing, age, gender or even the first names provide cues and constraints on the identity. Markov Random Fields have been shown to be a powerful tool to model both the fusion of different modalities as well as the dependencies between faces and persons in connected images.

In this chapter, we extend previous work on naming faces in TV series/multimedia data to naming persons. That is, we localize persons independently from faces with the goal to increase the recall on the named person instances in the video beyond those where a face is visible and detected. Although person detection is a difficult problem in itself and the recall of current approaches is not comparable with face detection methods, we show that with a current state-of-the-art detector we are able to increase both the number of named persons as well as the precision on the existing face tracks.

Following the work in the context of personal photo collections, we propose a late fusion scheme for identities in videos, modeling person and face tracks as random variables over identities in a Markov random field. As such, we can both incorporate different modalities into the recognition as well as incorporate constraints on the identities. We further propose to include information from transcripts and subtitles to indicate the presence of a person within a shot. So far, this information has only been used during training of face models and was not enforced during identification. Inference of the identities is performed jointly within a shot, in contrast to a track-by-track decision as in previous approaches on character naming.

For describing clothing, we employ RGB color histograms due to their simplicity. Although many more sophisticated features have been described in the literature, simple color histograms were shown to provide already an effective description of the general appearance of a person. We propose an automatic approach to assign identity labels to clothing clusters to avoid requiring manual labels. In contrast to previous work, our clothing descriptor is not dependent on the availability of faces.

Acknowledgment This chapter contains joint work with Makarand Tapaswi, who I supervised for his Master thesis (Tapaswi, 2011).

4.2. Preprocessing

In contrast to the previous chapter, we will describe the preprocessing steps on the data before going into details on the method. This simplifies the description of the method as it directly builds on the different parts of the pre-processing.

In the following, we will first discuss the localization of *persons* in the videos and the association with their possible face tracks. We will also briefly describe the computation of the clothing descriptor and the detection of scene boundaries in order to detect possible clothing change time points.

4.2.1. Person detection and tracking

Similar to faces, we first need to localize persons before reasoning about their identities. By and large, person detection and tracking is very similar to face detection and tracking, and all advantages of detector-based tracking similarly apply to person tracking. However, in contrast to faces, persons are very non-rigid “objects” due to their many degrees of freedom of how to move arms, legs and modify their body posture. This makes the detection step considerably harder and state-of-the-art person detection methods (*e.g.*, Bourdev and Malik (2009)) do not perform quite as well as their face detection counterparts.

Similar to faces, a multitude of detection methods is described in the literature. Commonly well perform methods based on histograms of oriented gradients (HOG), such as the original HOG-based holistic SVM approach (Dalal and Triggs, 2005), an extension to a part-based model with a latent SVM (Felzenszwalb et al., 2009) or an ensemble of hundreds of *Poselets* (Bourdev and Malik, 2009). Part-based models can deal to some extent with partly occluded persons which are very common in multimedia data due

Ep.	MOTA	MOTP	FP(R)	MISS(R)	BMM(R)	GMM(R)	TR	TP
1	0.670	0.658	25 (0.022)	248 (0.226)	2 (0.001)	87 (0.079)	0.806	0.964
2	0.626	0.629	16 (0.015)	314 (0.306)	0 (0.000)	53 (0.051)	0.747	0.970
3	0.645	0.653	50 (0.053)	241 (0.256)	0 (0.000)	42 (0.044)	0.764	0.913
4	0.592	0.623	43 (0.043)	285 (0.287)	1 (0.001)	76 (0.076)	0.751	0.929
5	0.575	0.619	51 (0.054)	284 (0.300)	0 (0.000)	66 (0.069)	0.739	0.918
6	0.518	0.512	100 (0.077)	399 (0.308)	1 (0.000)	123 (0.095)	0.729	0.875

Table 4.1.: Person tracking evaluation for BBT. The MOTA is consistently worse than for faces. However, this is mostly due to a high miss rate (MISSR), as expected due to the more difficult problem of person detection. Despite the high miss rate, we still obtain a track recall (TR) between 70% and 80%. There are almost no bad mismatches (BMM), *i.e.* track switches between different persons, which is important for later identification.

to crops at the image border. From these options, we select the Poselet approach by Bourdev and Malik (2009) due to its state-of-the-art performance.

For tracking, we follow (Huang et al., 2008) and associate detections from neighboring frames according to an affinity measure based on distance and size of adjoining person detections. In order to assess the performance of our tracker, we annotated person bounding boxes for every 10th frame of the first 6 episodes of *The Big Bang Theory*. The evaluation in terms of MOTA and related measures can be found in Table 4.1.

4.2.2. Clothing descriptor extraction

For clothing description, we use simple RGB color histograms at a fixed relative region within the person bounding box. Despite more advanced methods for clothing description, color histograms were shown to perform reasonably well in previous work. Due to their simplicity, they can be used without the need for other complex prerequisites such as a body pose estimator.

We compute one descriptor per person bounding box as determined by the tracker. We define the descriptor region as $(x, y, w, h) = (0.1 \cdot w, 0.2 \cdot h, 0.8 \cdot w, 0.3 \cdot h)$ within the person bounding box. The RGB histogram comprises $4 \times 4 \times 4$ bins, resulting in a 64-dimensional descriptor **c**. We normalize the descriptor to unit-norm, *i.e.* $c \leftarrow c/||c||$ To reduce the influence of single outlier bins, we threshold each bin at 0.1 and then re-normalize the descriptor, *i.e.* $c_i \leftarrow \min(0.1, c_i); c \leftarrow c/||c||$, similar to (Dalal and Triggs, 2005; Lowe, 2004).

We regard this color feature as a good baseline in order to determine what can be achieved with a very simple feature. Naturally, a more sophisticated descriptor is expected to improve overall identification performance. Similarly, a better segmentation of the clothing region, possibly in multiple sub-regions and jointly performed for all detections in the image should decrease the influence of background and occluders.

4.2.3. Scene detection

Characters can change their clothing multiple time within one episode or movie. This usually happens when the plot jumps in time or the location changes. Often, especially in sitcoms, such cuts are emphasized by a special audio jingle or visual sequence.

In BBT, special computer graphics-rendered sequences are used. They have a distinct color gradient as background. For simplicity, we employ a specialized method to detect these scenes. A color gradient is difficult to capture in a few colors. We therefore compute the 8 dominant colors for each frame in *Lab* color space according to (ISO/IEC 15938-3, 2001). We then backproject each pixel in the frame to its closest dominant color and compute the mean difference between the original frame and its dominant color backprojection

$$\mathrm{DCD} = \frac{1}{N}\sum_{x,y} ||I(x,y) - DCBP(x,y)|| \tag{4.1}$$

as the dominant color descriptor (DCD) for that frame. A simple threshold suffices to locate the special sequences of BBT without errors.

The scene detection results in a set of timestamps $\{t_i\}$ at which we assume the possibility of clothing changes.

4.3. Global identity model

In order to obtain identities for person tracks, we first need to build clothing models using the color features described above. We will describe an automatic method to associate some of the clothing descriptors with identities in the following. We will also describe how subtitles can be used as another modality next to clothing and faces for identification. Finally, we present in this section a model for jointly assigning identities within a shot, incorporating the different modalities in the identity decision.

4.3.1. Person clustering and identity assignment

In order to automatically assign identities to some of the person tracks, we first associate person tracks with face tracks. We transfer the face tracks' identities as obtained from Chapter 3 to their associated person tracks. Since face identification results are expected to be noisy (with accuracies as low as 60%), we cluster similar clothing descriptors and assign identities to clusters instead of individual descriptors to reduce the influence of noise from the labels. Also, outliers such as from occlusions can be overturned by other labels in the cluster.

1. Clustering of clothing descriptors We perform agglomerative clustering with Ward linkage (Ward, 1963) on the descriptors. The goal of the clustering of descriptors is to obtain pure clusters, *i.e.* in the ideal case we do not want to mix descriptors from different persons. Firstly, this is achieved by clustering descriptors within each scene separately, using the timestamps of scene boundaries from the scene detection (see Sec. 4.2.3). That is, clothing descriptor clusters are never merged across scenes. This prevents that the descriptors of two person get merged when they wear similar clothing albeit in different scenes. Secondly, we employ a low threshold θ_c for cluster merging.

See Table 4.2 for clustering results on BBT. The low threshold leads to an over-clustering of the descriptors, *i.e.* we usually obtain more clusters than characters in each scene. At the same time, we are able to maintain a high purity of the clusters. The number of clusters is significantly lower than the number of face tracks, indicating that we cluster across and not just within tracks.

2. Cluster identity assignment Given the clothing clusters, we now assign identities to some of the clusters using the identities from the face tracks.

We first start by associating face tracks with person tracks based on their relative location. We define an expected face region within the person bounding box as $(x, y, w, h) = (0.1 \cdot w, 0.0 \cdot h, 0.8 \cdot w, 0.2 \cdot h)$. If the face lies strictly within the expected face region for at least 5 frames, we associate face and person track, *i.e.* assume that they stem from the

Ep.	#scenes	#tracks	#clusters	avg. cluster size	avg. cluster purity
1	6	792	205	223	0.934
2	7	717	331	109	0.977
3	7	712	316	115	0.990
4	10	729	358	101	0.925
5	8	714	305	115	0.968
6	6	1157	282	163	0.937

Table 4.2.: Clustering results for BBT. Due to the over-clustering the cluster purity is very high.

same person. We will reuse the association also later in the joint identification (Sec. 4.3.3) to infer one joint identity for each associated face and track.

Using the associations, we accumulate identities for each cluster. For cluster $C_i = \{c_{ij}\}$ and associated face tracks identities $y_{ij} \in \{1..K, \varnothing\}$ ($\varnothing$ denotes for the *unknown* class), we compute the frame-based identity distribution as

$$P_{C_i}(y = k) = \frac{1}{|C_i|} \sum_j \mathbb{1}[y_{ij} = k] \quad , \tag{4.2}$$

i.e. we count how many frames in the cluster are associated with face tracks of each identity.

Since neither the clusters are completely pure, nor the face identities are fully correct, the accumulated identities can also be impure. We therefore impose two restrictions on assigning an identity to a cluster. First, at least 10% of the frames of the cluster must be assigned an identity, *i.e.* $|\{y_i j \neq \varnothing\}| > 0.1|C_i|$. Second, the most frequent identity must be assigned to at least 60% of the assigned frames, *i.e.* $\max_k P_{C_i}(y = k) > 0.6$. If these two conditions are met, we call the cluster *assigned*.

The likelihoods of the identities will further be used for obtaining track identity likelihoods (see below).

3. Track identity assignment In order to compute identity likelihoods for a whole track, we first compute identity likelihoods for all frames of the track. For a frame which already belongs to an assigned cluster, we directly use the corresponding cluster's identity likelihoods .

For all other frames $F_u = \{f_{ui}\}$, we compute the distance of the corresponding clothing descriptor to the assigned clusters' mean descriptors $\overline{c}_i = \frac{1}{|C_i|}\sum_j c_{ij}$. The frame's identity likelihoods are computed as a weighted sum over all assigned clusters' identity likelihoods

$$P_{f_u}(y=k) = \sum_i w_i \cdot P_{C_i}(y=k) \tag{4.3}$$

where the weight $w_i = \exp\{-(c_u - \overline{c}_i)/(\sum_j c_u - \overline{c}_i)\}$ controls the contribution of cluster C_i with an exponential decay based on the distance between the clothing descriptor and the respective cluster mean.

Given all individual frame likelihoods, we can compute the identity likelihoods of track t_i by averaging over its frames f_j

$$P_{t_i}(y=k) = \frac{1}{N}\sum_j^N P_{f_j}(y=k) \quad . \tag{4.4}$$

4.3.2. Speaker presence

We can usually assume that the current speaker is present in the frame. Given an identity of the current speaker (*e.g.*, from speech recognition or subtitles), we can therefore infer that at least one of the face or person tracks should have the same identity. We already exploited this assumption in the generation of weak labels for training face models (see Sec. 3.3.2). Since there can be multiple persons at the same time in the frame, we cannot directly associate the speaker identity with one specific track. In Sec. 3.3.2 we detected whether any of the faces was speaking by a separate mechanism. However, when the face of the current speaker is not detected, *i.e.* we only have a person track, or the speaker detection fails due to a non-frontal pose, we cannot perform the association. Therefore, we integrate *presence* as a independent cue, not associated to any track. The association will be performed implicitly during the identification (see Sec. 4.3.3).

One possible way to determine the current speaker would be to perform speaker recognition using the audio channel of the video. However, this would require a separate training step and introduce new errors. Instead, we use the subtitle-transcript alignment as described in Sec. 3.3.2 to obtain the identity of the current speaker, achieving a very high precision of 99.9% correctly labeled subtitles with a recall of 91.2%.

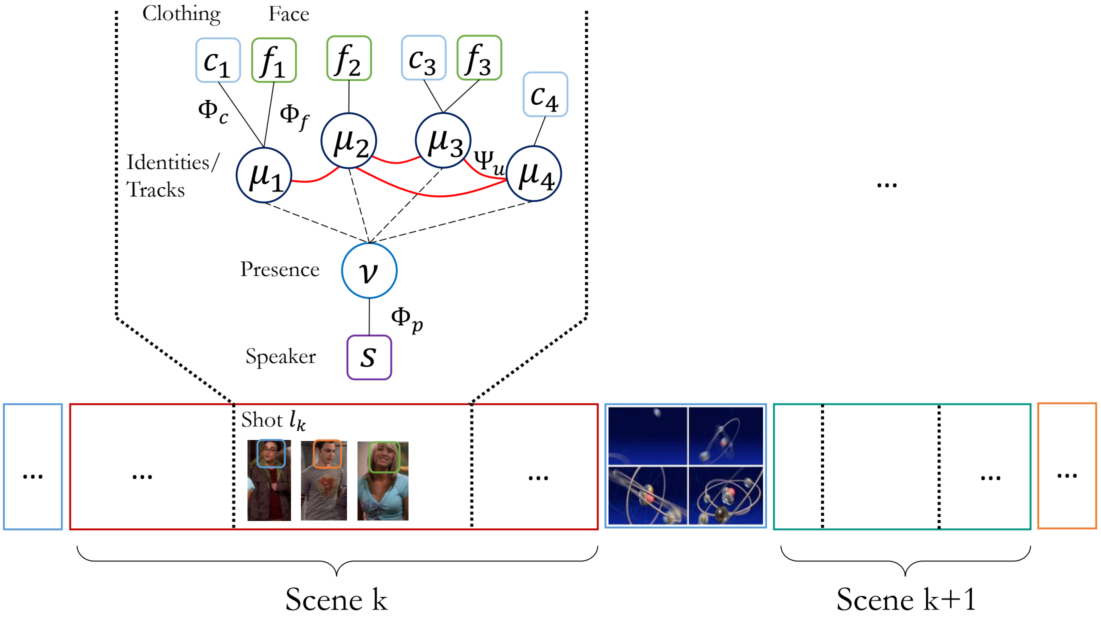


Figure 4.1.: Illustration of the MRF for fusion and joint identification. The video is divided into scenes and shots. For each track, there is one associated identity variable μ_i. Each identity variable is associated with respective face results f_i and clothing results c_i, if present, and the speaker presence s via the joint presence variable ν. Overlapping tracks are further interconnected via the uniqueness potential Ψ_u.

4.3.3. Fusion of modalities and joint identification

A *Markov random field* (MRF) describes the joint distribution of a set of random variables, where dependencies between random variables are given by edges within the graph. In this framework, we model face/person tracks as random variables over identities and introduce connections between them within shots, for example to (softly) enforce uniqueness of identity assignments of co-occurring tracks or the presence of the current speaker.

Let for each shot denote $\mu = \{\mu_1, \ldots, \mu_n\}$ the set of identity variables associated with a face or person track. We encode the identity variables as $k+1$-dimensional vectors, for the k *known* person plus the *unknown* class. Each of the identity variables is associated with the respective face results f_i and clothing results c_i, if available (see Fig. 4.1). If further per-track modalities are available, *e.g.* gender, age or similar, they can be integrated straightforwardly into the model.

We define the joint density in terms of an energy $E(\mu)$ as

$$P(Y=\mu)=\frac{1}{Z}e^{-E(\mu)} \quad , \tag{4.5}$$

where $Z=\int_{\mu} e^{-E(\mu)}$ is the partition function, ensuring that $\int_{\mu} P(Y=\mu)=1$. The energy $E(\mu)$ consist of unary terms $\Phi_k(\mu_i)$ for the individual modalities and pairwise terms $\Psi_l(\mu_i,\mu_j)$ for the relationships between different variables

$$E(\mu)=\sum_k w_k \sum_i \Phi_k(y_i)+\sum_l w_l \sum_{i,j} \Psi_l(y_i,y_j) \quad . \tag{4.6}$$

We are interested in finding the identity assignment $\mu*$ that minimizes $E(\mu)$, corresponding to a maximization of $P(Y=\mu)$. We disregard Z in the following since it is constant in μ and therefore only a constant factor in the maximization of $P(Y=\mu)$.

In the following, we will motivate and define the unary and pairwise terms for the different modalities and relationships. Intuitively, a unary term should be low when the identity variable matches the likelihoods from the respective modality and high if they do not match.

Face unary We define the face unary as

$$\Phi_f(\mu_\mathrm{i})=-\langle \mu_\mathrm{i},\mathbf{f}_\mathrm{i}\rangle \quad . \tag{4.7}$$

The negative dot product has the desired property of being low, when the variable matches the face likelihoods. For example, if f_{i1} is high, indicating that the face matched the model of person 1, $\Phi_f(\mu_\mathrm{i})$ will be low if the identity variable μ_{i1} is also high and vice versa.

Clothing unary The clothing unary is basically identical to the face unary, just taking into account the identity likelihoods from the clothing-based recognition

$$\Phi_c(\mu_\mathrm{i})=-\langle \mu_\mathrm{i},\mathbf{c}_\mathrm{i}\rangle \quad . \tag{4.8}$$

Presence unary In order to reason about the presence of characters, we need to combine the individual identity variables to a shot-wide presence variable ν. We deduce the soft presence ν of the characters by

$$\nu = \sigma\left(\sum_k \mu_{\mathrm{k}}\right) \quad , \tag{4.9}$$

where $\sigma(x) = (1 + \exp(-x))^{-1}$ is the sigmoid function. The sigmoid ensures that the presence term cannot be dominated by one character if multiple tracks of the same character are in the same shot, *e.g.* due to occlusions.

The speaker unary associates the speaker presence result s with the presence variable ν as

$$\Phi_p(\nu) = \langle 1 - \nu, s \rangle \quad , \tag{4.10}$$

penalizing if the character should be present according to s, but is not according to ν.

Uniqueness constraint Let $P^{(-)}$ be the set of track pairs that co-occur, *i.e.* share at least one common frame. For co-occurring tracks, we define a pairwise term between the corresponding identity variables μ_i and μ_j that penalizes the assignment of the same identity

$$\Psi_u(\mu_i, \mu_j) = \langle \mu_{i,1:k}, \mu_{j,1:k} \rangle \tag{4.11}$$

where $\mu_{*,1:k}$ is the reduced identity variable excluding the *unknown* class. Since we do not resolve unknown tracks further, we do not enforce a unique assignment on unknown.

Regularization In order to avoid that μ grows unbounded, we regularize μ using a standard L2 regularization term. Following the above notation, the regularization term is

$$\Phi_r(\mu) = \sum_i \langle \mu_i, \mu_i \rangle \; (= \|\mu\|^2) \tag{4.12}$$

Given the above unaries and pairwise terms, the complete energy becomes

$$\begin{aligned} E(\mu) = w_f \sum_i \Phi_f(\mu_i) + w_c \sum_i \Phi_c(\mu_i) + \dots \\ + w_p \Phi_p(\nu) + w_u \sum_{(i,j) \in P^{(-)}} \Psi_u(\mu_i, \mu_j) + w_r \Phi_r(\mu) \end{aligned} \tag{4.13}$$

with respective weights w_* to control the relative influence of each modality. For our following experiment, we use equal weights $w_f = w_c = w_p = w_r = 1$ for all but the uniqueness term, which we enforce slightly more with $w_u = 2$.

Energy minimization In order to jointly assign identities using the above model, we minimize Eq. 4.13

$$\mu* = \arg\min_{\mu} E(\mu) \quad \text{such that} \quad 0 \leq \mu_{ij} \forall i, j \quad . \tag{4.14}$$

The positivity constraint on μ ensures that μ maintains positive scores. We minimize Eq. 4.13 using an active set method (Nocedal and Wright, 2006) as implemented in MATLAB.

The obtained $\mu*$ induces the identity labeling on each of the corresponding tracks: the identity for track i is assigned as

$$k* = \operatorname{argmax}_k \mu_{ik} \quad . \tag{4.15}$$

4.4. Evaluation

For evaluation, we employ the same data set as in the previous chapters. Statistics on the face tracks of the data set can be found in Table 3.1. We conduct experiments using different underlying tracks for the identity variables μ_i, namely face tracks and a combination of face and person tracks.

Our first motivating thesis was that by taking into account multiple modalities, we can improve the face recognition performance by resolving ambiguities in the face descriptors by a different modality. We therefore start by instantiating one μ_i for each face track, and use the information from clothing, speaker and uniqueness to improve *face* recognition results.

	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	Avg.
Face only	89.50	90.24	76.36	78.79	81.30	65.92	80.35
MRF F+C	93.30	93.17	80.76	80.75	83.21	67.45	83.11
MRF F+C+U	93.61	92.85	81.36	81.73	**84.73**	**70.62**	84.15
MRF F+C+U+S	**94.52**	**94.80**	**82.73**	**82.54**	84.54	70.27	**84.90**

Table 4.3.: Joint identification and multimodal fusion results for BBT on *face* tracks. Track-level accuracy increases consistently when adding clothing (C), uniqueness (U) and speaker presence (S) terms into the MRF energy.

However, the coverage of the above is not increased over face-only identification. We argued that a true person identification approach should go beyond faces only. We therefore instantiate one μ_i for each track in the union of face and person tracks, *i.e.* all associated face and person tracks *and* all singular tracks, both face and person tracks, which could not be associated. In this second setting, we associate face and clothing unaries only to those identity variables, where the respective track is present.

Multimodal face recognition We start by investigating the influence of the different modalities on the *face* recognition performance. That is, how much can the incorporation of clothing, speech and uniqueness help improve face recognition performance?

We instantiate one μ_i per face track and associate the different modalities to their respective identity variable according to the face-person track association. We evaluate the incorporation of different modalities step by step (see Tables 4.3 and 4.6 for BBT and Table 4.9 for BUFFY), namely *face + clothing* (F+C), *face + clothing + uniqueness* (F+C+U) and *face + clothing + uniqueness + speaker presence* (F+C+U+S). At each step, track-level accuracy, frame-level accuracy and correct classification rate increase consistently, improving face recognition performance by an absolute 4% in average (track-level) accuracy for BBT. For BUFFY, a similar improvement can be observed, although less strong with an improvement of only about 2%, possibly due to problems of reliably distinguishing clothing colors in dark image conditions.

Despite the clothing models being learned using the results of the face recognition, they alone (without uniqueness or speaker) already provide an accuracy improvement of about 2.7% for BBT. The improvement from the speaker modality for BBT is small (about 0.75% compared to 1.6% for BUFFY) despite its high precision (*cf*. Sec 4.3.2). One reason for this is that for some shots there is no face track of the current speaker

	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	Avg.
Using automatic face recognition results (*cf*., Chapter 3)							
Face only	71.28	66.98	58.51	60.34	57.54	45.64	60.05
Face only + prior	79.92	73.47	69.33	66.80	66.23	56.75	68.75
MRF F+C	91.36	87.53	78.09	75.10	76.96	64.10	78.86
MRF F+C+U	91.89	88.06	79.25	76.15	**79.28**	**68.21**	80.47
MRF F+C+U+S	**92.55**	**90.05**	**79.90**	**77.47**	78.99	67.61	**81.09**
Using ground truth face labels							
Face only	79.65	74.54	77.06	76.68	70.43	68.12	74.41
Face only + prior	88.30	81.03	87.89	83.14	79.13	79.23	83.12
MRF F+C	**95.88**	94.69	**97.81**	91.17	93.91	88.46	93.65
MRF F+C+U	95.74	96.15	97.16	**93.41**	**95.94**	**91.03**	94.91
MRF F+C+U+S	**95.88**	**96.55**	97.29	**93.41**	95.51	90.94	**94.93**

Table 4.4.: Joint identification and multimodal fusion results for BBT on the union of face and person tracks (see text). Track-level accuracy increases consistently when adding clothing (C), uniqueness (U) and speaker presence (S) terms into the MRF energy. Using ground truth face labels instead of automatic face recognition results, we test the limit of our approach and reach almost 95% track level accuracy.

due to a tracker failure. In such shots, the presence modeling not only does not improve results, but possibly also overturns an otherwise correct decision to comply with the presence of the speaker. In BUFFY on the other hand, there are many close-up shots with only a single face present (the current speaker), whose identification benefits from the hint on the current speaker identity.

The uniqueness constraint provides a significant reduction of the false acceptance rate, causing to overturn one or more identically labeled tracks from a known identity to unknown. Clothing and speaker modalities on the other hand do not provide improvements in FAR.

Extending recognition to person tracks We now extend the task to naming tracks stemming from both faces and persons. We construct joint tracks from the face-person-track association as described in Sec. 4.3.1 (2. *Cluster identity assignment*), *i.e.* we combine all associated face and person tracks to a single joint track. Face and person tracks that are not associated with any other track are considered their own “joint” track.

	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6	Avg.
Using automatic face recognition results (*cf*., Chapter 3)							
Face only	61.34	52.25	57.67	54.40	53.79	54.04	55.58
Face only + prior	68.22	57.75	62.69	58.72	62.18	58.66	61.37
MRF F+C	75.71	64.19	67.25	68.27	66.90	62.19	67.42
MRF F+C+U	76.32	64.57	66.85	69.60	67.71	62.70	67.96
MRF F+C+U+S	**77.63**	**65.74**	**68.39**	**70.93**	**69.58**	**64.43**	**69.45**
Using ground truth face labels							
Face only	77.73	73.26	76.56	70.60	72.97	77.71	74.80
Face only + prior	84.62	78.76	81.58	74.92	81.36	82.32	80.59
MRF F+C	93.32	87.29	91.02	88.70	90.63	88.74	89.95
MRF F+C+U	94.03	**88.45**	92.50	90.53	**91.70**	88.53	90.96
MRF F+C+U+S	**94.33**	**88.45**	**92.83**	**90.78**	91.35	**88.82**	**91.09**

Table 4.5.: Joint identification and multimodal fusion results for BUFFY on the union of face and person tracks (see text).

We first compute a face recognition-based baseline for these joint tracks. To this end, we assign any joint track its corresponding face recognition result, if a face track is associated with the joint track (denoted *Face only* in Tables 4.4 and 4.5). For all other tracks, *i.e.* those that only stem from a person track, we can without further information assign the *max-prior* identity, *i.e.* the most likely identity for a track in the 6 episodes for each series. For BBT, the max-prior identity is *Leonard*. The respective results are denoted *Face only + prior* in the result table. Compared to the results on face tracks alone, the face recognition-based baselines on joint tracks perform worse in terms of accuracy due to the higher number of joint tracks and the low accuracy of the *max-prior* assignment of only ≈ 30%.

Again, we evaluate the incorporation of different modalities step by step. Adding clothing provides a big jump in performance, improving results by 10% to 78% over face-only recognition for BBT and by 6% to 67% on BUFFY. In this setting, the clothing modality has more impact than in the previous experiment, since it provides the strongest cue for the person tracks without a face.

See Tables 4.4 and 4.5 for an overview of the performance in terms of track-level accuracy and Tables 4.7 and 4.10 for detailed comparison.

Ground truth face labels The joint recognition depends on face identification results on two levels. They are first used to bootstrap the clothing models and then integrated as one modality in the joint identification. Since face recognition is a very active field of research, we expect that new advances will improve face recognition results, for example due to better descriptors or better matching between different face poses.

In order to explore the limit of our approach in dependency of the face recognition results, we assume face recognition results to be 100% accurate, but leave everything else as before. We perform identification on joint face-person tracks, see Tables 4.4 and 4.5 for an overview on track accuracies and Tables 4.8 and 4.11 for a detailed comparison. When considering only the ground truth face results, we reach about 83% accuracy, indicating that almost one fifth of the tracks cannot be identified via faces. By incorporating clothing, uniqueness and speaker presence we obtain a track-level accuracy of about 94%.

	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	Avg.
Track-level Accuracy							
Face only	89.50	90.24	76.36	78.79	81.30	65.92	80.35
MRF F+C	93.30	93.17	80.76	80.75	83.21	67.45	83.11
MRF F+C+U	93.61	92.85	81.36	81.73	**84.73**	**70.62**	84.15
MRF F+C+U+S	**94.52**	**94.80**	**82.73**	**82.54**	84.54	70.27	**84.90**
Frame-level Accuracy							
Face only	94.35	94.70	80.80	87.35	84.93	77.27	86.57
MRF F+C	94.89	96.05	84.46	88.86	86.10	78.33	88.11
MRF F+C+U	94.91	95.67	85.24	**89.72**	**87.45**	79.50	88.75
MRF F+C+U+S	**95.34**	**96.79**	**85.83**	89.37	87.39	**79.60**	**89.05**
Correct Classification Rate							
Face only	94.91	94.89	91.31	90.69	91.39	85.99	91.53
MRF F+C	95.36	96.24	**94.70**	91.84	93.37	**87.16**	**93.11**
MRF F+C+U	95.38	95.86	92.98	**92.43**	**93.65**	84.91	92.54
MRF F+C+U+S	**95.82**	**96.98**	93.46	92.07	93.59	85.22	92.86
False Acceptance Rate							
Face only	39.62	**100.00**	82.43	98.97	79.70	73.88	79.10
MRF F+C	**33.63**	**100.00**	79.05	88.27	86.92	75.80	77.28
MRF F+C+U	**33.63**	**100.00**	62.75	**80.30**	**74.90**	**53.64**	67.53
MRF F+C+U+S	**33.63**	**100.00**	**61.54**	**80.30**	**74.90**	54.82	**67.53**
False Rejection Rate							
Face only	1.54	2.60	2.99	0.08	2.26	0.64	1.68
MRF F+C	**0.66**	**1.26**	**2.20**	**0.00**	**1.34**	**0.39**	**0.98**
MRF F+C+U	0.86	2.25	3.47	0.16	1.49	4.56	2.13
MRF F+C+U+S	0.86	1.41	3.08	0.60	1.49	4.45	1.98

Table 4.6.: Joint identification and multimodal fusion results for BBT on *face* tracks. This table extends Table 4.3 with more details. Track-level accuracy, frame-level accuracy and correct classification rate increase consistently when adding clothing (C), uniqueness (U) and speaker presence (S) terms into the MRF energy. Adding the uniqueness constraint significantly improves the false acceptance rate.

	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	Avg.
Track-level Accuracy							
Face only	71.28	66.98	58.51	60.34	57.54	45.64	60.05
Face only + prior	79.92	73.47	69.33	66.80	66.23	56.75	68.75
MRF F+C	91.36	87.53	78.09	75.10	76.96	64.10	78.86
MRF F+C+U	91.89	88.06	79.25	76.15	**79.28**	**68.21**	80.47
MRF F+C+U+S	**92.55**	**90.05**	**79.90**	**77.47**	78.99	67.61	**81.09**
Frame-level Accuracy							
Face only	85.04	83.86	75.19	77.52	75.59	66.87	77.35
Face only + prior	88.19	86.79	79.05	80.07	79.35	71.17	80.77
MRF F+C	94.67	93.07	83.82	84.93	83.75	74.07	85.72
MRF F+C+U	94.63	93.93	84.43	85.49	**85.12**	**76.12**	86.62
MRF F+C+U+S	**94.94**	**95.29**	**84.66**	**86.09**	84.93	75.70	**86.94**
Correct Classification Rate							
Face only	85.77	84.09	84.47	82.76	83.18	74.97	82.54
Face only + prior	88.98	87.02	88.93	85.49	87.40	80.04	86.31
MRF F+C	95.44	93.32	**93.37**	90.22	92.97	**83.84**	**91.52**
MRF F+C+U	95.39	94.18	91.77	89.68	93.45	81.95	91.07
MRF F+C+U+S	**95.71**	**95.54**	91.99	**90.31**	**93.46**	81.67	91.45
False Acceptance Rate							
Face only	53.75	**100.00**	85.14	97.64	85.93	78.29	83.46
Face only + prior	53.75	**100.00**	85.14	97.64	85.93	78.29	83.46
MRF F+C	**45.91**	**100.00**	78.26	91.06	90.96	80.40	81.10
MRF F+C+U	**45.91**	**100.00**	63.26	74.61	**82.41**	**56.38**	**70.43**
MRF F+C+U+S	**45.91**	**100.00**	**62.96**	**74.37**	84.21	57.62	70.84
False Rejection Rate							
Face only	2.05	2.96	3.00	**0.07**	2.01	1.03	1.85
Face only + prior	2.05	2.96	3.00	**0.07**	2.01	1.03	1.85
MRF F+C	**0.58**	**1.62**	**1.95**	0.33	**1.15**	**0.50**	**1.02**
MRF F+C+U	0.87	2.67	2.97	1.53	1.28	5.48	2.47
MRF F+C+U+S	0.84	1.69	2.75	1.51	1.18	5.93	2.32

Table 4.7.: Joint identification and multimodal fusion results for BBT on the union of face and person tracks (see text), using *automatic* face recognition results. This table extends Table 4.4 with more details.

	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	Avg.
			Track-level Accuracy				
Face only	79.65	74.54	77.06	76.68	70.43	68.12	74.41
Face only + prior	88.30	81.03	87.89	83.14	79.13	79.23	83.12
MRF F+C	**95.88**	94.69	**97.81**	91.17	93.91	88.46	93.65
MRF F+C+U	95.74	96.15	97.16	**93.41**	**95.94**	**91.03**	94.91
MRF F+C+U+S	**95.88**	**96.55**	97.29	**93.41**	95.51	90.94	**94.93**
			Frame-level Accuracy				
Face only	91.17	89.62	92.36	89.34	88.06	87.61	89.69
Face only + prior	94.32	92.55	96.22	91.89	91.81	91.91	93.12
MRF F+C	**97.40**	97.47	**99.46**	95.38	97.53	**94.41**	96.94
MRF F+C+U	97.22	98.53	98.97	96.33	**98.65**	93.93	97.27
MRF F+C+U+S	97.25	**98.70**	99.18	**96.37**	98.62	94.03	**97.36**
			Correct Classification Rate				
Face only	91.42	89.66	93.12	91.94	90.56	89.80	91.08
Face only + prior	94.63	92.60	97.57	94.67	94.78	94.87	94.85
MRF F+C	**97.65**	97.53	**99.63**	**97.78**	98.82	**96.98**	**98.06**
MRF F+C+U	97.30	98.53	99.05	97.37	**99.04**	94.56	97.64
MRF F+C+U+S	97.33	**98.69**	99.29	97.45	99.03	94.72	97.75
			False Acceptance Rate				
Face only	22.06	25.47	12.54	48.06	32.29	24.59	27.50
Face only + prior	22.06	25.47	12.54	48.06	32.29	24.59	27.50
MRF F+C	15.90	25.47	1.59	39.01	12.96	19.91	19.14
MRF F+C+U	**6.83**	**0.00**	**1.52**	**18.63**	**4.54**	**9.59**	**6.85**
MRF F+C+U+S	**6.83**	**0.00**	**1.52**	19.15	4.72	9.84	7.01
			False Rejection Rate				
Face only	**0.00**	**0.00**	0.39	**0.00**	**0.54**	0.99	0.32
Face only + prior	**0.00**	**0.00**	0.39	**0.00**	**0.54**	0.99	0.32
MRF F+C	**0.00**	0.01	**0.07**	0.33	0.60	**0.23**	**0.21**
MRF F+C+U	0.49	0.57	0.73	1.19	0.75	3.10	1.14
MRF F+C+U+S	0.58	0.41	0.49	1.09	0.75	3.03	1.06

Table 4.8.: Joint identification and multimodal fusion results for BBT on the union of face and person tracks (see text), using *ground truth* face labels. This table extends Table 4.4 with more details.

	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6	Avg.
Track-level Accuracy							
Face only	78.52	71.12	**75.46**	75.78	73.69	69.65	74.04
MRF F+C	79.52	72.11	74.96	76.56	75.60	69.65	74.73
MRF F+C+U	79.52	71.91	73.62	76.00	75.48	69.39	74.32
MRF F+C+U+S	**80.90**	**73.61**	75.21	**77.11**	**78.21**	**70.72**	**75.96**
Frame-level Accuracy							
Face only	84.18	78.66	**80.57**	83.26	82.85	74.53	80.67
MRF F+C	84.93	78.85	79.40	**83.66**	83.69	74.79	80.89
MRF F+C+U	84.88	79.10	78.61	83.02	83.56	74.55	80.62
MRF F+C+U+S	**86.52**	**80.30**	79.73	83.63	**85.19**	**75.62**	**81.83**
Correct Classification Rate							
Face only	84.35	84.67	**80.57**	85.79	87.46	77.53	83.39
MRF F+C	85.09	85.45	79.34	**86.20**	88.53	77.83	83.74
MRF F+C+U	85.04	84.92	78.54	85.48	88.38	77.58	83.33
MRF F+C+U+S	**86.67**	**86.04**	79.68	86.11	**89.47**	**78.70**	**84.44**
False Acceptance Rate							
Face only	59.69	77.09	19.87	100.00	65.49	**94.32**	69.41
MRF F+C	59.69	83.31	**13.17**	100.00	67.02	94.89	69.68
MRF F+C+U	59.69	75.65	**13.17**	**98.16**	67.02	94.89	68.10
MRF F+C+U+S	**53.40**	**73.70**	**13.17**	**98.16**	**59.75**	94.89	**65.51**
False Rejection Rate							
Face only	0.80	0.95	**0.53**	0.46	0.03	**0.81**	0.60
MRF F+C	0.88	**0.87**	0.81	**0.14**	0.03	**0.81**	**0.59**
MRF F+C+U	0.88	1.50	1.43	0.68	0.03	1.37	0.98
MRF F+C+U+S	**0.29**	1.21	1.73	0.68	**0.02**	1.43	0.89

Table 4.9.: Joint identification and multimodal fusion results for BUFFY on *face* tracks.

	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6	Avg.
Track-level Accuracy							
Face only	61.34	52.25	57.67	54.40	53.79	54.04	55.58
Face only + prior	68.22	57.75	62.69	58.72	62.18	58.66	61.37
MRF F+C	75.71	64.19	67.25	68.27	66.90	62.19	67.42
MRF F+C+U	76.32	64.57	66.85	69.60	67.71	62.70	67.96
MRF F+C+U+S	**77.63**	**65.74**	**68.39**	**70.93**	**69.58**	**64.43**	**69.45**
Frame-level Accuracy							
Face only	77.37	70.76	71.89	74.36	74.31	68.37	72.84
Face only + prior	78.52	72.28	72.94	75.78	76.40	70.16	74.35
MRF F+C	83.83	75.69	75.05	79.63	78.16	71.04	77.23
MRF F+C+U	84.01	76.28	74.87	80.25	78.82	70.87	77.52
MRF F+C+U+S	**85.25**	**77.32**	**75.99**	**80.47**	**79.84**	**72.18**	**78.51**
Correct Classification Rate							
Face only	77.64	77.50	72.45	78.15	81.70	71.61	76.51
Face only + prior	78.80	79.22	73.52	79.63	84.09	73.49	78.13
MRF F+C	84.08	83.79	75.65	83.69	86.23	74.48	81.32
MRF F+C+U	84.25	82.91	75.13	83.71	86.18	74.17	81.06
MRF F+C+U+S	**85.47**	**83.90**	**76.27**	**83.98**	**86.71**	**75.55**	**81.98**
False Acceptance Rate							
Face only	74.19	81.90	61.46	100.00	78.32	93.85	81.62
Face only + prior	74.19	81.90	61.46	100.00	78.32	93.85	81.62
MRF F+C	62.26	87.61	61.19	100.00	79.27	95.16	80.91
MRF F+C+U	60.65	75.50	**40.18**	**87.74**	73.59	**92.65**	71.72
MRF F+C+U+S	**56.77**	**74.06**	**40.18**	88.33	**69.07**	**92.65**	**70.18**
False Rejection Rate							
Face only	**0.71**	0.85	**0.53**	0.20	**0.03**	**0.73**	**0.51**
Face only + prior	**0.71**	0.85	**0.53**	0.20	**0.03**	**0.73**	**0.51**
MRF F+C	0.78	**0.78**	0.92	**0.14**	**0.03**	0.76	0.57
MRF F+C+U	1.32	1.70	1.89	1.63	0.15	2.31	1.50
MRF F+C+U+S	0.78	1.41	1.88	1.97	0.10	2.36	1.42

Table 4.10.: Joint identification and multimodal fusion results for BUFFY on the union of face and person tracks (see text), using *automatic* face recognition results. This table extends Table 4.5 with more details.

	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6	Avg.
Track-level Accuracy							
Face only	77.73	73.26	76.56	70.60	72.97	77.71	74.80
Face only + prior	84.62	78.76	81.58	74.92	81.36	82.32	80.59
MRF F+C	93.32	87.29	91.02	88.70	90.63	88.74	89.95
MRF F+C+U	94.03	**88.45**	92.50	90.53	**91.70**	88.53	90.96
MRF F+C+U+S	**94.33**	**88.45**	**92.83**	**90.78**	91.35	**88.82**	**91.09**
Frame-level Accuracy							
Face only	92.74	90.54	90.21	88.93	90.49	92.16	90.84
Face only + prior	93.90	92.06	91.26	90.35	92.58	93.95	92.35
MRF F+C	97.32	95.49	94.93	95.40	95.40	**95.22**	95.63
MRF F+C+U	97.40	**95.57**	96.31	96.09	**95.78**	94.01	95.86
MRF F+C+U+S	**97.54**	95.43	**96.45**	**96.39**	95.72	94.35	**95.98**
Correct Classification Rate							
Face only	92.91	91.98	90.98	90.57	93.95	92.80	92.20
Face only + prior	94.07	93.69	92.05	92.06	96.33	94.68	93.81
MRF F+C	97.44	**96.88**	95.75	96.62	**97.27**	**95.79**	**96.62**
MRF F+C+U	97.51	95.92	96.96	96.88	97.18	94.46	96.49
MRF F+C+U+S	**97.66**	95.79	**96.97**	**97.21**	97.21	94.83	96.61
False Acceptance Rate							
Face only	37.42	20.69	55.89	43.25	34.15	20.22	35.27
Face only + prior	37.42	20.69	55.89	43.25	34.15	20.22	35.27
MRF F+C	25.48	15.35	53.70	28.60	17.87	15.58	26.10
MRF F+C+U	**23.87**	**7.10**	42.83	**19.40**	**14.21**	**14.64**	20.34
MRF F+C+U+S	25.48	7.42	**34.61**	19.77	14.87	14.89	**19.51**
False Rejection Rate							
Face only	**0.00**	**0.23**	**0.00**	**0.00**	0.32	**0.15**	**0.12**
Face only + prior	**0.00**	**0.23**	**0.00**	**0.00**	0.32	**0.15**	**0.12**
MRF F+C	**0.00**	0.85	0.07	0.06	**0.16**	0.35	0.25
MRF F+C+U	0.18	2.22	0.18	0.62	0.85	2.00	1.01
MRF F+C+U+S	0.18	2.09	0.24	0.55	0.85	1.93	0.97

Table 4.11.: Joint identification and multimodal fusion results for BUFFY on the union of face and person tracks (see text), using *ground truth* face labels. This table extends Table 4.5 with more details.

Chapter 5

A Time Pooled Track Kernel

In Chapter 3 we learned frame-based face models. That is, our training data consisted of descriptors of individual frames, and for identification we first classified each frame separately before fusing the individual results for a joint track decision.

However, the number of frames to consider during training can quickly grow too large to handle. Video data in the multimedia domain alone amounts to millions of hours of data, with hundreds of hours added per day. Efficiency, both in terms of computational and memory requirements, should therefore be taken into account. Especially for kernel-based methods, the memory requirements grow quadratically with the number of training features: the Gram matrix consists of the value of the kernel function for every pair of features.

However, tracks can also be regarded as image sets, which can be advantageous for multiple reasons: Image sets can be represented more compactly than the set of individual frames (*e.g.*, Hu et al. (2011)). Especially for distance or Gram matrices, memory requirements reduce to the order of $O(M^2)$, where M is the number of *sets*, compared to $O(N^2)$ with N denoting the number of *descriptors* across all sets combined (usually $M \ll N$). This is especially important in the context of large scale learning, where it might be infeasible to keep all individual descriptors (let alone a kernel's Gram matrix with memory requirement $O(N^2)$) in memory. Therefore, given a finite amount of memory, track representations can have more discriminative power than (a subset of) individual frames. Furthermore, at test time, image set comparisons and decisions can be more efficient to compute, possibly at the expense of a one-time pre-computational step. This is desirable when quick iterations of training and testing are required, *e.g.* when user feedback is obtained and training cycles are performed with a human in the loop (see Sec. 5.3).

In this chapter, we present a generic time-based pooling kernel for tracks.

5.1. Background and related work

For large scale learning, different advances have been made recently. Shalev-Shwartz et al. (2007) proposed a primal linear Support Vector Machine solver based on stochastic gradient descent that scales well to large problem instances. For some classes of non-linear kernels, mappings to approximate feature maps $\hat{\Psi}(x)$ have been proposed in order to benefit from the speed-ups for linear SVMs (Maji et al., 2008; Vedaldi and Zisserman, 2012). However, these techniques usually rely on an explicit feature map expansion which is impractical for very large or even infinite dimensional feature maps.

Different specific set distances have been devised for image sets in general. We can differentiate between how approaches represent an image set, *e.g.* via its covariance matrix (Wang et al., 2012) or its convex hull (Hu et al., 2011), and the way image sets are compared, such as smallest distance between subspaces (Cevikalp and Triggs, 2010; Hu et al., 2011) or correlation-based measures (Wang et al., 2008).

Kernels on sets are more prevalent in the context of local features, where an image or object is represented by a set of local features (*e.g.*, Wallraven et al. (2003)). Robustness for such set kernels can for example be improved by non-uniform weighting (Lyu, 2005). The pooled NBNN kernel (Rematas et al., 2012) pools base kernel comparisons on local features over sub-classes or visual-word-like clusters. This is closely related to our approach, however, it requires clustering in feature space which can be computationally expensive, whereas we do not.

In the context of face recognition multimedia data, Parkhi et al. (2014) proposed a feature composition method to obtain one feature per track based on Fisher vector encoding of dense SIFT features. Other encodings such as the covariance matrix of the descriptors (Wang et al., 2012), can be easily implemented for face tracks as well.

5.1.1. Discussion and contribution

We propose a time-based pooling kernel for tracks which incorporates as special cases both the normalized sum kernel (Lyu, 2005) and frame-wise base kernels. The kernel pools local kernel evaluations over time, leveraging the structure of tracks. In contrast to the pooled NBNN kernel, this is very efficient since it does not require a clustering step in feature space, which can be computationally expensive.

With the proposed time pooling kernel we both reduce the memory requirements during training and are able to speed up training-testing iterations at the same time. Due to the structure of the kernel, classification/identification of *all tracks* can be performed simultaneously and efficiently by means of a single matrix multiplication. Quick turn-around times (training a new model and inferring new identities for all tracks) allows efficient incorporation of feedback into the learning process.

5.2. Time pooled kernels

We are interested in applying convex optimization methods to classification of time-based sets of features (*e.g.*, face or person tracks). As already briefly discussed in Sec. 3.2.1, many learning methods refer in their respective loss functions only within dot products $\langle \mathbf{x}, \mathbf{y} \rangle$ to the training data. In order allow non-linear decision boundaries, one can replace the dot product by a kernel function $k(\mathbf{x}, \mathbf{y})$. This corresponds to computing the dot product between the features in a different (usually higher dimensional) feature space: $k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$.

We already introduced multinomial logistic regression in Sec. 3.2.1 as one such classifier which can be extended by a kernel function to non-linear decision boundaries, resulting in the objective function

$$\min_{\mathbf{w}} \frac{\lambda}{2} ||\mathbf{w}||^2 - \frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{M} \mathbb{1}[y_i = c] \ln \left(\frac{e^{f_{\mathbf{w}_\mathrm{c}}(\mathbf{x}_\mathrm{i})}}{\sum_\mathrm{z} e^{f_{\mathbf{w}_\mathrm{z}}(\mathbf{x}_\mathrm{i})}} \right) \quad . \tag{5.1}$$

Another popular classifier, the (two-class) Support Vector Machine (SVM) can be extended in the same way:

$$\min_{\mathbf{w}} \frac{1}{2} ||\mathbf{w}||^2 + \frac{C}{N} \sum_{i=1}^{N} \max\{0, 1 - y_i f_{\mathbf{w}}(\mathbf{x}_\mathrm{i})\} \quad , \tag{5.2}$$

replacing the dot product $\langle \mathbf{w}, \mathbf{x}_\mathrm{i} \rangle$ by a function $f_\mathrm{w}(\mathbf{x}) = \sum_j w_j k(\mathbf{x}_\mathrm{j}, \mathbf{x})$.

Kernels are commonly defined on vectors since the relation to the dot product is evident. However, kernels can also be defined on other structures, *e.g.* strings (Lodhi et al., 2002) or vector sets (Kondor and Jebara, 2003), thus making such structures available as input for above convex optimization schemes. In this chapter, we propose and evaluate a family of set kernels for *tracks*.

5.2.1. Implementation of learning with kernels

In this section, we want to briefly motivate from an implementation perspective why the reduction in the number of entities from frames to tracks is beneficial.

Let $Q \in \mathbb{R}^{N \times N}$ be a kernel's Gram matrix with $Q_{ij} = k(\mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{j})$ for all pairs of training features, N being the number of training features. We can then write the replacement $f_\mathrm{w}(\mathbf{x}_\mathrm{i})$ for the dot product in the loss functions as

$$f_\mathrm{w}(\mathbf{x}_\mathrm{i}) = \sum_j w_j k(\mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{j}) = [\mathbf{Q}]_i \mathbf{w} \quad , \tag{5.3}$$

where $[\mathbf{Q}]_i$ is the ith row of $\mathbf{Q}$. In both the SVM's as well as MLR's loss function we need to compute $f_\mathrm{w}(\mathbf{x}_\mathrm{i})$ for all x_i due to the summation over all training features. We can thus compute

$$\mathbf{v} = \mathbf{Q}\mathbf{w} \tag{5.4}$$

with $v_i = f_\mathrm{w}(\mathbf{x}_\mathrm{i})$. For both SVMs and MLR, the Gram matrix therefore appears in a matrix multiplication with the parameter vector.

If the number of training samples is large, it becomes infeasible to store $\mathbf{Q}$ in memory. There are different approaches to deal with this problem. For example, one can reduce the kernel bases x_i to a subset of size $M < N$, leading to a non-square Gram matrix $\mathbf{Q} \in \mathbb{R}^{M \times N}$. Similarly, one can also reduce the number of training features, reducing the other dimension of $\mathbf{Q}$. For SVMs, a minimum solution for its loss (Eq. 5.2) is usually sparse (selecting a subset of the kernel basis as *support vectors*). Therefore, the optimization itself can be performed by sequentially adding support vectors without computing the full $\mathbf{Q}$ beforehand (*e.g.*, Platt (1998)). However, even when computing $\mathbf{Q}$ lazily, storing all computed entries of $\mathbf{Q}$ for later usage is still infeasible for large problems. Therefore, entries of $\mathbf{Q}$ are often recomputed multiple times, trading off computational time for memory requirement.

Our approach in this chapter also reduces the size of $\mathbf{Q}$, however not by subsampling but by choice of a different kernel function. By operating on tracks instead of features, the size of the full Gram matrix $\mathbf{Q}$ reduces from $N \times N$ (N being the number of training features) to $M \times M$, with M being the number of tracks ($M \ll N$).

5.2.2. Pooling over time

Let $k(\mathbf{x}, \mathbf{y}) = \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle$ denote a *base* or *local kernel* defined on individual frames (respectively their descriptors) $\mathbf{x} \in X$ and $\mathbf{y} \in Y$. Instead of using individual local kernel evaluations directly, we pool their values over time with a pooling function $\Phi(\cdot)$, which for example can be the max or average $\frac{1}{MN}\sum\sum$ over the respective local kernel values of a set of feature pairs. Let further X and Y be time-consecutive sets of features, *e.g.* features extracted from tracks. We define a *track kernel* as

$$\begin{aligned} K\Big(X, Y\Big) &= K\Big(X, Y; k(\cdot,\cdot), \Phi(\cdot)\Big) \\ &= \Phi\Big(\{k(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in X, \mathbf{y} \in Y\}\Big) \quad . \end{aligned} \tag{5.5}$$

This construction spans a family of different kernels, depending on the choice of local kernel and pooling function. $K(X, Y)$ is a Mercer kernel (*i.e.*, positive semi-definite, p.s.d.) if the local kernel $k(\cdot,\cdot)$ is a Mercer kernel itself *and* the pooling operation is from a set of operations (*e.g.*, sum) that preserve positive semi-definiteness (Lyu, 2004).

For example, using the RBF kernel $k_{RBF}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x}-\mathbf{y}||^2}{2\sigma^2}\right)$ as local kernel and $\frac{1}{MN}\sum\sum$ as pooling operation results in

$$K(X, Y) = \frac{1}{|X||Y|}\sum_{x \in X}\sum_{y \in Y} \exp\left(-\frac{||\mathbf{x}-\mathbf{y}||^2}{2\sigma^2}\right) , \tag{5.6}$$

which corresponds to the normalized sum kernel (Lyu, 2005) with a RBF kernel as base kernel. Since $k_{RBF}(\mathbf{x}, \mathbf{y})$ is a Mercer kernel and $\sum$ is a valid construction operation, Eq. 5.6 is a Mercer kernel.

In contrast to Lyu (2005) and Rematas et al. (2012), the pooling of base kernels over time does not require any a-priori clustering or nearest-neighbor search in feature space and

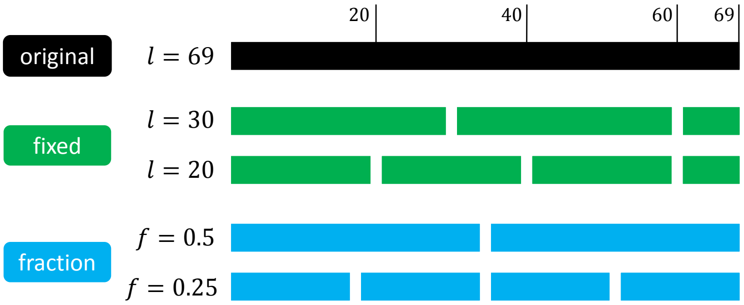


Figure 5.1.: Different examples for splitting a track (black) into fixed-time (green) and fraction-time (blue) subtracks.

thus is very efficient. This is based again on the assumption that all features of one track belong to the same class/person, as we already assumed in previous chapters.

A drawback of the normalized sum kernel is that as it possibly averages out few positive correspondences between the features of two tracks by many other negative correspondences. This could be for example the case when persons turn their face/head within one track which usually amounts to changes in the extracted features due to the different poses.

We approach this problem from two perspectives. First, we employ non-averaging pooling operations which can deal with many negative correspondences, *e.g.* max or its variant $\frac{1}{N}\sum \max_N$, where we average over the N highest local kernel results. Second, we reduce such variations by applying the kernel not on full tracks, but on sub-tracks and thus improve the coherence of the features. Our goal is similar to Rematas et al. (2012), however, the pooling is performed over time. The shorter the pooling period is, the more coherent the sets are. In the extreme case this can go down to single frames, which we will discuss in Sec. 5.2.3.

Fixed-time splitting A first way of defining the pooling length is by splitting tracks into equal-length time-continuous sets of features. Let X be the original set of features for one track, then $X_i^{fixed(l)}$ are the new subtracks of equal length l, combining the ith set of l consecutive features of the track. The last subtrack can be shorter when the original track length $len(X)$ is not a multiple of l. In most cases fixed-time splitting avoids a bias by track length.

Fraction-time splitting One possible issue with the fixed-time splitting is that long tracks get over-represented in the new subtrack sets. Instead of splitting into fixed-length sets, we can split each track into equal-size fractions $f \in (0,1]$, *i.e.* construct $1/f$ subtracks $X_i^{frac(f)}$ from one track, each with $len(X_i^{frac(f)}) = f \cdot len(X)$. With fraction-time splitting the relative number of tracks of each class is preserved. A visualization of the two different track splitting variants can be found in Fig. 5.1.

5.2.3. Special cases and relation to other methods

Normalized sum kernel and extensions As already discussed above, the normalized sum kernel is included in our family of track kernels as a special case (*cf*. Eq. 5.6) with pooling operation $\frac{1}{MN}\sum\sum$ and no further splitting of tracks ($f = 1$). With the power-kernel $k(\cdot,\cdot)^p$ we obtain the soft-max Mercer kernel of (Lyu, 2005).

Single-frame classification By splitting all tracks into individual frames (*i.e.*, $l = 1$), we arrive at frame-wise classification as we used in Chapter 3. When employing $\frac{1}{MN}\sum\sum$ as pooling operation, track kernel-based and frame-wise classification have commonalities even for $l > 1$. In the frame-wise case for MLR-based classification, results are averaged over the frames of the test-track (at test-time, *cf*. Eq. 3.20):

$$c^* = \arg\max_c \frac{1}{|T|} \sum_i^{|T|} \sum_m^{W_F} w_m^{(c)} k(\mathbf{x}_i, \mathbf{x}_m) \tag{5.7}$$

where $|T|$ is the track length, W_F the number of frame-based kernel bases (individual features) from the training data, and $w_m^{(c)}$ the model parameters learned by minimization of Eq. 5.1.

For the track kernel-based classification (with $\frac{1}{MN}\sum\sum$ pooling) we compute for track T at test time

$$c^* = \arg\max_c \sum_m^{W_T} w_m^{(c)} K(T, T_m) \quad . \tag{5.8}$$

Expanding $K(\cdot,\cdot)$ to the summation over local kernels $k(\cdot,\cdot)$, we obtain

$$c^* = \arg\max_c \sum_m^{W_T} w_m^{(c)} \frac{1}{|T||T_m|} \sum_i^{|T|} \sum_j^{|T_m|} k(\mathbf{x}_i, \mathbf{x}_j^m) \quad . \tag{5.9}$$

By reordering the sums,

$$c^* = \arg\max_c \frac{1}{|T|} \sum_i^{|T|} \sum_m^{W_T} \frac{w_m^{(c)}}{|T_m|} \sum_j^{|T_m|} k(\mathbf{x}_i, \mathbf{x}_j^m) \tag{5.10}$$

we obtain a similar structure to Eq. 5.7.

For $l = 1$ each pooling T_m set only contains a single frame, *i.e.* $|T_m| = 1$. Setting $|T_m| = 1$ in Eq. 5.10, the similarity to Eq. 5.7 is apparent.

For $l > 1$, all frames in a pooling set T_m share the parameter $w_m^{(c)}$, whereas in Eq. 5.7 there is one $w_m^{(c)}$ for each frame of the kernel basis. This parameter sharing comes with a significant reduction in the number of kernel bases, *i.e.* $W_T << W_F$.

Training/testing speed-up Since pooling over time is performed within the kernel, we can pre-compute significant parts of Eq. 5.10 required for both training (*cf*. Eq. 5.1) *and* testing. In the above example, the summation term over T and T_m can be pre-computed and stored once for all track combinations. The equivalence to Eq. 5.8 reveals that this is exactly the Gram matrix of the track kernel. Due to the reduction in kernel bases ($W_T << W_F$), it is feasible to store the complete Gram matrix without resorting to subsampling or approximations: the Gram matrix for 10000 tracks ($\sim$ 1 season of an average 20min TV series) fits well in current-sized main memory ($10000^2 \cdot 8Byte \approx 750MB$).

Due to the pre-computation, we also avoid recomputing base kernel evaluations and summations over individual frames at test time (compared to the single frame case, see Eq. 5.7). The combination of the reduction in the size of the Gram matrix and a simpler classification results in a speed-up for subsequent training/testing iterations. A similar pre-computation for the base kernel evaluations could also be done for the single-frame case, but the much larger number of instances (frames instead of tracks) prohibits this for all practical instances.

5.2.4. Time and space complexity

Training The computation of the kernel's Gram matrix requires of $O(N^2)$ local kernel evaluations, where N is the number of features. Therefore, the computation of the full track kernel Gram matrix is in $O(N^2 \cdot k(d))$, with $k(d)$ being the complexity of the local kernel evaluation depending on feature dimensionality d. The memory requirements for the full Gram matrix depend on the pooling factors f or l, reducing the required memory compared to the frame-wise kernel by l^2, resulting in $O(\frac{N^2}{l^2})$. This allows, as argued before, to pre-compute and store the Gram matrix for multiple rounds of training (*cf*. Sec. 5.3).

Testing At test time, we benefit from the fact that pooling over each (sub-)track was already performed at training time. In the case of MLR, this reduces classification to a single matrix multiplication $\mathbf{Qw}$ and obtaining the maximum over rows:

$$c_i^* = \arg\max_c \frac{e^{[\mathbf{Q}]_i \mathbf{w}^{(c)}}}{\sum_z e^{[\mathbf{Q}]_i \mathbf{w}^{(z)}}} = \arg\max_c [\mathbf{Qw}]_i \quad , \tag{5.11}$$

where $\mathbf{Q} \in \mathbb{R}^{N/l \times N/l}$ is the Gram matrix of the track kernel, $[\mathbf{Q}]_i$ the ith row of $\mathbf{Q}$, $|C|$ the number of classes and $w \in \mathbb{R}^{N/l \times |C|}$ the parameter vector obtained by minimizing the MLR loss function (Eq. 5.1). Thus, testing is dominated by the matrix multiplication $\mathbf{Qw}$ and results in time complexity of $O(\frac{N^2|C|}{l^2})$.

5.3. Learning with a human in the loop

As an application of the track kernel, we consider training models for identification with a human in the loop, combining the strengths of an automatic classifier and a human operator. A possible motivation might be to prepare the identities of all tracks of a TV series for release on a streaming platform. In that case, all tracks should be associated with the correct identity, corresponding to a recognition accuracy of 100%, which is improbable to accomplish with a current automatic approach. A related application can be found in safety and security scenarios, where for example human operators are supported by the learning algorithm to reconstruct the path of a thieve in a shopping mall, learning new models while receiving feedback from the operator.

Given a fixed classifier (*e.g.*, trained in an automatic way using labels from subtitle-transcript matching), the naïve solution would be to let the classifier identify all tracks once and then let a human correct all wrongly identified tracks manually. However, assuming that the classifier is able to generalize, new training data will result in a better classifier that is capable of identifying additional test data correctly outside of the new training data. This makes it beneficial to re-run training of the classifier and testing all tracks once a few incorrect samples have been corrected. This avoids wasting expensive labeling time by automatically correcting some tracks which would otherwise have had to be corrected manually.

This raises the questions of a) which training data to label, and b) how many new training samples S to label. The literature on *active learning* (*e.g.*, Settles (2010)) mostly deals with the former question, while number of samples $|S|$ is usually set to 1 for one iteration, disregarding the training and inference time required. Batch mode active learning usually deals with *which samples* to select when $|S| > 1$. We are rather interested in the relationship between training/inference time and the *number* of samples to label each round to minimize the time to reach 100% recognition accuracy.

Our simple model of required labeling time is as follows. Let t_{pre} be a fixed amount of time needed to set up a classifier and pre-compute the kernel's Gram matrix. Let t_{init} be the time it takes to label one training sample before any learning has been performed, t_{fb} the feedback time on one wrongly classified sample, and t_{train} and t_{test} the time it takes to re-train the classifier and re-test all test samples, respectively. Further, let N be a number of labeled samples in each round. The total labeling time can then be computed as follows

$$\begin{aligned} t_{total} = {} & t_{pre} + N_{init} \cdot t_{init} \\ & + k \cdot \max\{t_{train} + t_{test}, N \cdot t_{fb}\} \end{aligned} \tag{5.12}$$

To minimize unnecessary manual labeling, we should label $\hat{N}$ tracks in each round, such that $\hat{N} \cdot t_{fb} \approx t_{train} + t_{test}$. As discussed in Sec. 5.2.3, we can achieve fast training and inference with the track kernel, since we move the most time consuming step of kernel computation to t_{pre}. Therefore, $\hat{N}$ can be very small, reducing the amount of unnecessary labeling to a minimum.

BBT			BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	Avg.
single frame MLR			89.50	90.24	76.36	78.79	81.30	65.92	80.35
track-repr CDL (Wang et al., 2012)			80.52	72.36	67.27	64.11	62.79	48.41	65.91
pool	split	Mercer							
$\frac{1}{MN}\sum\sum$	$f = 1$	✓	90.26	88.29	78.79	75.53	81.11	70.86	80.81
$\frac{1}{MN}\sum\sum$	$f = 0.5$	✓	90.11	88.62	80.30	76.18	82.44	71.09	81.46
$\frac{1}{MN}\sum\sum$	$l = 30$	✓	93.15	90.41	80.61	78.96	86.83	73.09	83.84
max max	$f = 1$		89.65	90.08	76.36	78.47	81.87	67.92	80.73
max max	$f = 0.5$		93.46	92.36	81.67	78.63	88.55	74.15	84.80
max max	$l = 30$		92.54	91.71	80.45	76.35	82.44	70.51	82.33
$\frac{1}{N}\sum\max_{1..N}$	$f = 1$		93.91	93.17	82.73	78.79	88.55	71.80	84.83
$\frac{1}{N}\sum\max_{1..N}$	$f = 0.5$		89.35	88.62	76.52	74.39	80.15	71.92	80.16
$\frac{1}{N}\sum\max_{1..N}$	$l = 30$		93.76	93.01	85.45	78.79	87.79	73.09	85.32

BUFFY			BF-1	BF-2	BF-3	BF-4	BF-5	BF-6	Avg.
single frame MLR			78.52	71.12	75.46	75.78	73.69	69.65	74.04
track-repr CDL (Wang et al., 2012)			65.83	54.38	56.03	62.44	59.52	51.64	58.31
pool	split	Mercer							
$\frac{1}{MN}\sum\sum$	$f = 1$	✓	71.86	59.86	66.75	67.56	65.36	57.59	64.83
$\frac{1}{MN}\sum\sum$	$f = 0.5$	✓	73.74	64.94	69.01	71.89	67.62	63.18	68.40
$\frac{1}{MN}\sum\sum$	$l = 30$	✓	75.25	65.44	69.77	72.22	70.24	63.80	69.45
max max	$f = 1$		78.14	67.83	71.19	73.56	72.62	66.99	71.72
max max	$f = 0.5$		77.51	69.12	71.27	75.11	72.14	66.90	72.01
max max	$l = 30$		78.77	70.72	74.37	76.67	74.17	68.77	73.91
$\frac{1}{N}\sum\max_{1..N}$	$f = 1$		77.89	67.53	70.69	73.56	72.86	66.64	71.53
$\frac{1}{N}\sum\max_{1..N}$	$f = 0.5$		77.26	68.92	71.44	75.11	72.14	66.99	71.98
$\frac{1}{N}\sum\max_{1..N}$	$l = 30$		78.27	70.42	73.87	76.44	73.45	68.50	73.49

Table 5.1.: Baseline results are reported in the first two rows for BBT and BUFFY, respectively. CDL (Wang et al., 2012) is an example of a track-based representation method. The bottom section shows the performance of different instantiations of the time pooled track kernel. The max max and average-N-max (with $N = 5$) variants with a fixed-time splitting of $l = 30$ perform best on average, despite not being Mercer kernels. The normalized sum variants (rows 1-3 bottom section) perform worst among the different track kernel variants, however, better than CDL. For all variants, splitting tracks into sub-sets generally increases performance.

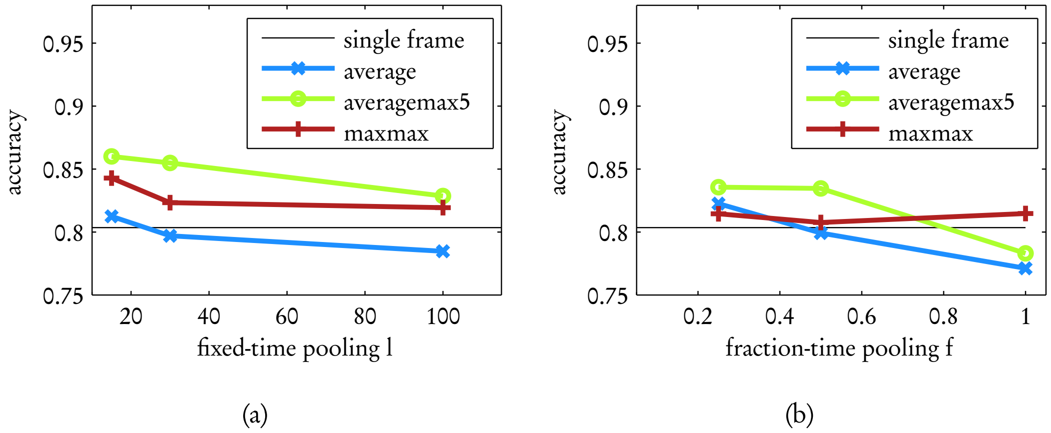


Figure 5.2.: Influence of the pooling parameters. Comparison of mean accuracy on BBT 1-6 with different pooling functions for (a) fixed-time pooling and (b) fraction-time pooling. The black line denotes the single frame-based recognition accuracy.

5.4. Evaluation

We perform experiments again on the data set of BBT and BUFFY episodes. We test different instantiations of track kernels. For a fair comparison, we use the same base kernel as in the baseline, a polynomial kernel of degree 2: $k(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}^T\mathbf{y})^2$. We evaluate pooling over the full tracks (*i.e.*, $f = 1$), fixed-time pooling with $l = 30$ (split each track into equal-length sub-tracks of size 30, which corresponds to roughly one half of the average track length) and fraction-pooling with $f = 0.5$ (split each track into exactly 2 sub-tracks). We also compare different basic pooling strategies, namely *normalized sum* ($\frac{1}{MN}\sum\sum k(\cdot,\cdot)$), *single maximum* ($\max\max k(\cdot,\cdot)$), and *average N-max*, an average of the maximum N base kernel values ($\frac{1}{N}\sum \max_{1..N} k(\cdot,\cdot)$). For the latter, we keep $N = 5$ over all experiments. The results of the comparison can be found in Tbl. 5.1.

Baseline As first baseline, we employ the MLR frame-based approach from Chapter 3. We further compare against Covariance Discriminative Learning (CDL) (Wang et al., 2012), which is an example of combining features of a track to a joint track-feature. For CDL, tracks are represented by their covariance matrix. We follow Wang et al. (2012) and add a small positive diagonal to the covariance matrix to ensure it is positive definite: $\mathbf{C}_X^* = cov(\mathbf{X}) + 10^{-3}\mathbf{I}$. The kernel between two tracks is defined as $k(\mathbf{X},\mathbf{Y}) = \left\|\log(\mathbf{C}_X^*)\cdot\log(\mathbf{C}_Y^*)\right\|_F$. Using this kernel, we train an MLR classifier (Eq. 5.1). CDL

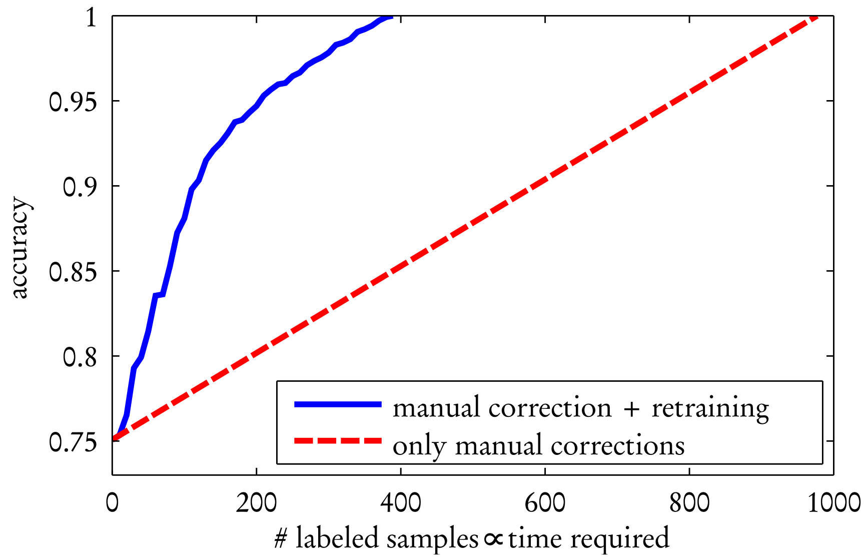


Figure 5.3.: Training with feedback on BBT 1-6. We retrain/test every 10 corrected samples. After correcting less than 400 samples, 100% accuracy is reached. Without re-training/testing almost 1000 samples would need to be manually corrected.

performs notably worse than the frame-based MLR approaches with 65.9% on BBT and 58.3% on BUFFY vs. 80.4% and 74%, respectively (see Tbl. 5.1).

Influence of splitting/time pooling For all variants of the track kernel, splitting tracks into sub-sets increases performance. Fig. 5.2 displays the influence of the two pooling parameters f and l for different pooling operations. We can see that for both fixed-time and fraction-time splitting a smaller subset size leads to higher recognition performance. This is not surprising since smaller subsets are more coherent and thus learning can select from more representative subsets. Both max-based pooling operations perform similarly. The normalized sum kernel (denoted as *average* in the plot legend) performs slightly worse, possibly owing to the imbalance of good and bad matching feature pairs for a given track pair.

Learning with feedback In Fig. 5.3, we compare the time required to manually label all tracks of BBT to 100% accuracy versus the time required when incorporating feedback with repeatedly retraining the classifier. Retraining *and* inference takes between 5 and 10 seconds on all 3920 tracks in our implementation (compared to minutes for the single-frame classifier). In our experience (from labeling many tracks of two TV series), it is possible to correct on average 1 wrongly classified track per second with an appropriate user interface (*i.e.*, displaying many tracks at once). We therefore set $N = 10$. There were 977 incorrectly classified tracks after the initial classifier run. We can see in Fig. 5.3 that by re-training and inferring new identities every 10 corrected samples, we reduce the amount of samples to correct to less than 400. This reduces the time to fully correct the initial recognition result to less than half.

Chapter 6

Conclusion

Face and person tracking and identification is a problem that arises in many automated video analysis tasks for automatic meta data generation and as a basis for higher level tasks. In this thesis, we have presented methods for robust face tracking, identifying faces and persons with additional data outside faces and integration of feedback into the learning process. While we have targeted only TV series as a data source in this thesis, the majority of the results are applicable to other data domains and applications. In the following, we summarize the contributions made in this work and outline possible directions for future work.

To track faces we proposed a detector-based multi-pose face tracker based on a particle filter. Different face poses are covered by a bank of detectors, each for a different head pose. By integrating the head pose in the track state, we can efficiently select a single detector for the observation model to avoid evaluating all 49 detectors for every particle. In addition, we obtained an estimate of the head pose and a rough configuration of facial landmarks, which can be used in subsequent analysis steps. We evaluated the tracker on a large data set of two diverse TV series. We have shown that tracking performance improves consistently by adding more out-of-plane detectors. In its complete configuration, our tracker improves MOTA on average by about 0.15 over a frontal-only detector. The tracker shows a higher track recall, *i.e.* finds more of the existing face appearance and also obtains longer tracks on average. For subsequent identification steps, both are important, as we can identify more faces, because they are found, and identify them better, because we have more samples per track to perform a robust decision. Without modification, the tracker has been applied successfully to other data domains as well, such as broadcast news, surveillance data or live webcam feeds.

We approached the task of person identification from two different perspectives. On the one hand, we proposed a joint learning framework for learning face models which integrates unlabeled data and constraints in addition to labeled data to learn better face models. Since labels and constraints can be obtained automatically, our method does not require any human supervision. We have shown that by integration of this additional data our method improves face recognition rates by an average 2%. On the other hand, we proposed a Markov random field-based fusion approach to integrate multiple cues into a joint identity decision. Since a face is not always visible for each person in the video, we extended the problem of naming faces in multimedia data to naming persons. In order to be able to identify persons, we explored clothing as a cue, again without requiring human supervision by leveraging face results to bootstrap clothing models. We demonstrated that the fusion and joint identification is beneficial for both identification of face tracks (improving results on average by 4%) and identification of joint face-person tracks (improving results by over 11%).

Finally, we proposed a novel family of kernels defined on tracks instead of individual frame-level features. The main motivation of the track kernel is a reduction of entities for training face and person models, resulting both in a reduction of memory requirements as well as a speedup during training and testing. We made use of the speedup to efficiently integrate human feedback in train-test iterations to increase the effectiveness of human feedback.

6.1. Limitations and future work

While we achieved a promising improvement in identification performance, there is room for further advancements.

Pose invariant face recognition In recent years, descriptors for faces have shifted towards local descriptors around landmarks for increased robustness and have shown impressive performance on standard face recognition data sets. We would expect that such descriptors provide an equivalent performance increase on multimedia data. A shortcoming of our approach is that we do not explicitly model the pose of the face during recognition. The employed affine 2d alignment is not sufficient to correct for the distortions induced by different view points of the 3-dimensional face. An interesting avenue of future work would therefore be to integrate learning pose-aware face models in combination with unlabeled data and constraints.

Clothing features Similar to the face descriptors, our employed descriptors for clothing are simple. In the context of clothing descriptions, different avenues of future work are thinkable. For one, a more fine-grained segmentation of the clothing region would promise to improve the discrimination between different persons. The larger the region for clothing descriptor extraction is, the more important is the handling of occlusions between different persons. One possibility to deal with occlusions could be for example to perform the segmentation of clothing jointly for all persons in a frame at the same time, taking into account the interdependencies of nearby persons.

Dependency on subtitles/transcripts The generation of automatic labels depends strongly on the availability of subtitles and transcripts. While subtitles are ubiquitously available, transcripts are usually supplied by fans and might not always be available. Some attempts have been made to obtain labeled tracks automatically without using transcripts. However, it might require a new level of understanding of the spoken text to reliably infer identities for all characters and in sufficient quantities directly from the text instead of annotated speech segments.

Unknown recognition and resolution One limitation of our proposed approach is the recognition of unknowns. We obtain unknown labels through the same mechanism as for the known characters (subtitle-transcript alignment + speaking face detection). However, the number of labels obtained in this way is low and covers only a few of the unknowns (many of the background characters never speak), and thus the recognition performance on unknowns is low. One possible option to resolve this problem would be to use an outside set of a large number of unknown face descriptors such as from the Labeled Faces in the Wild (LFW) data set to cover the space of unseen faces better.

Another interesting prospective would be to actually resolve unknowns (even those without training data) further into individual people instead of rejecting them jointly as one class. This would require for example some form of clustering to group similar unknown faces. However, judging from our experience in labeling ground truth unknown identities, such a task is even very hard for humans. Often this is due to unknowns only being visible in the background, thus influencing their face size and the camera focus.

Feedback for earlier stages We used face identities to bootstrap clothing models, which in turn improved face identity results. A possible option would be to iterate this further and use the new face identities to improve the original clothing models. Such a form of co-training has been explored in different domains and has provided promising results.

Generally, feedback from later stages in the pipeline could also be used to improve earlier stages. For example, one could improve the tracking by employing character specific face and clothing models to link individual tracklets or even find further localization of the characters which were previously missed by both the face and the person tracker.

Appendix A

Tracking Evaluation Plots for BUFFY

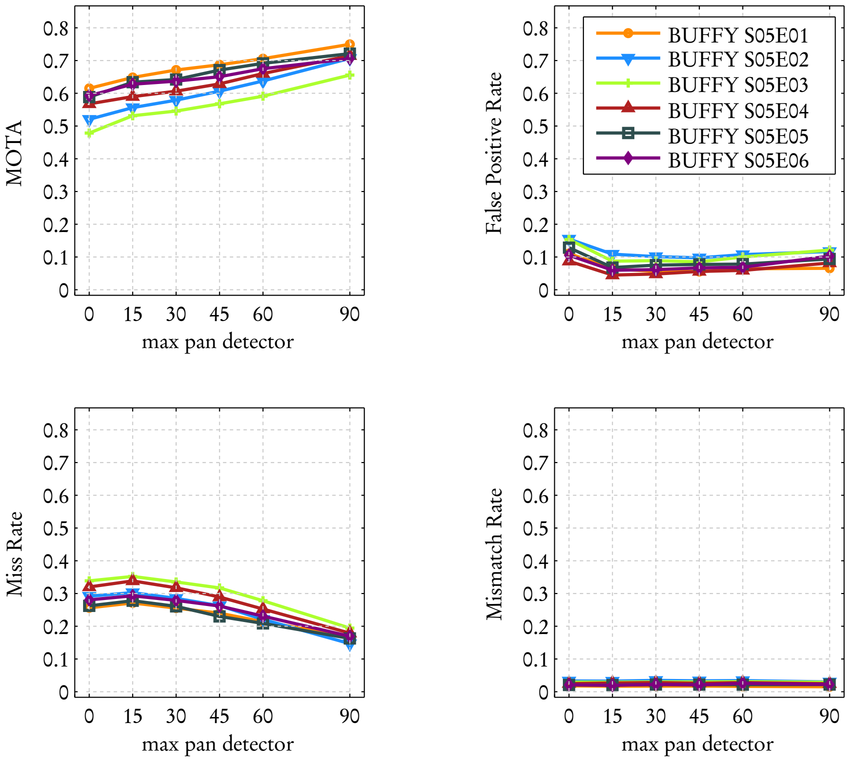


Figure A.1.: Tracking performance with increasingly out-of-plane rotated detectors on BUFFY. The x-axis denotes the maximum yaw angle of the underlying detectors, *e.g.*, at $x = 30$ the tracker uses detectors for yaw angles of 0, 15 and 30 degrees (inclusive). The corresponding plot for BBT can be found in Fig. 2.3.

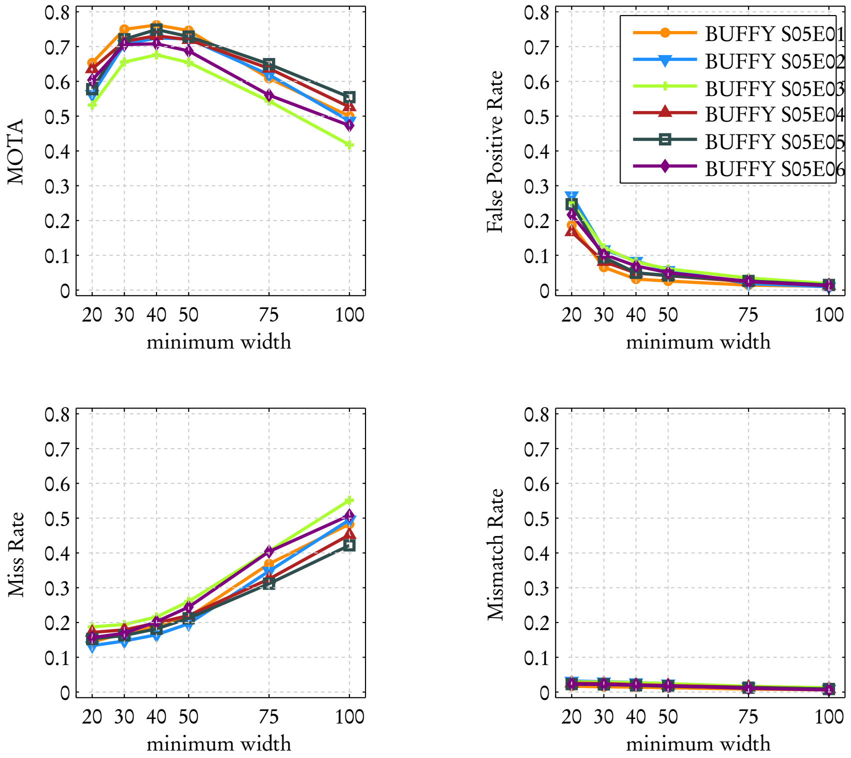
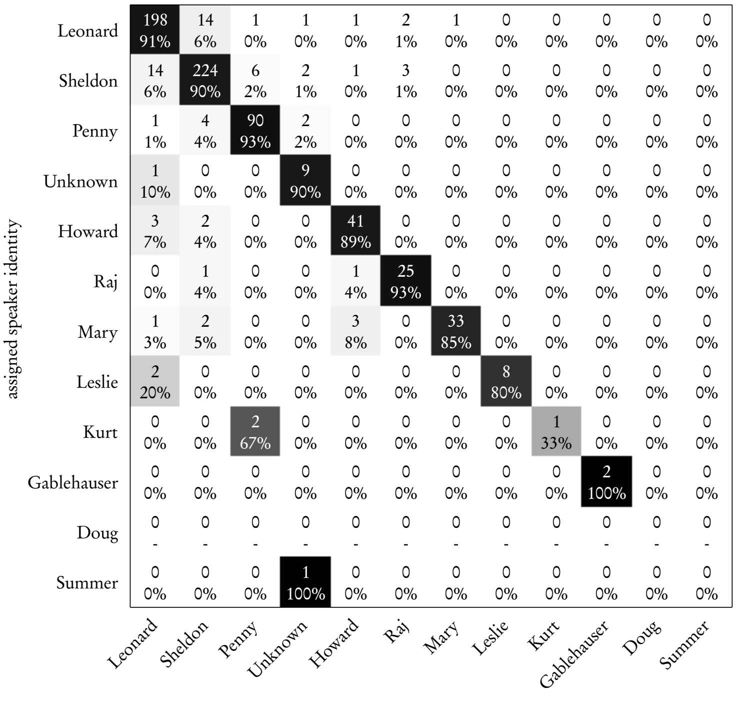


Figure A.2.: Tracking performance over increasing minimum face sizes. By rejecting tracks with average face width below 30px decreases false positive rate faster than miss rate increases and thus leads to an increase in MOTA.

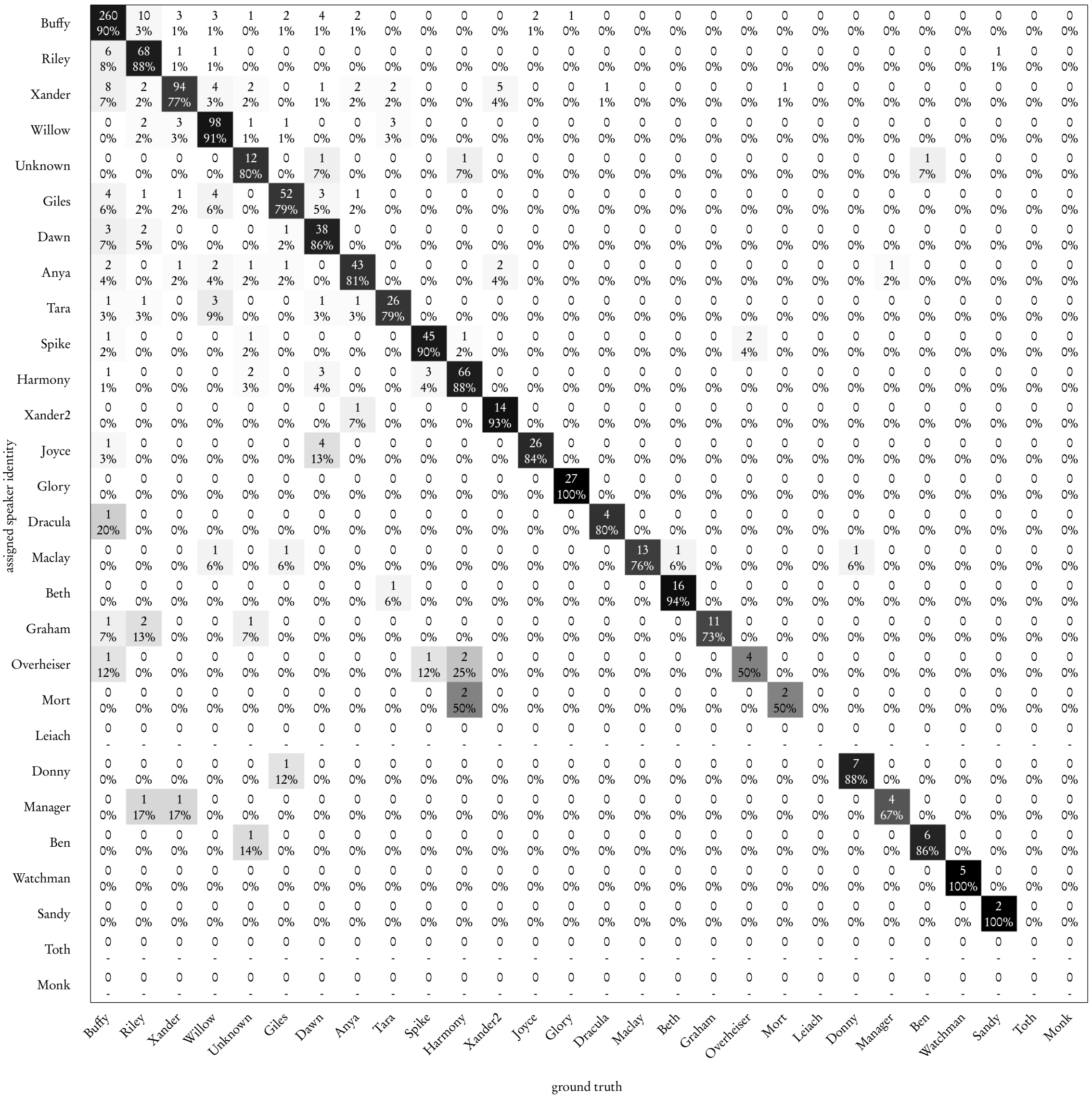
Appendix B

Automatic Speaker Assignment Confusion Matrices



assigned speaker identity \ ground truth	Leonard	Sheldon	Penny	Unknown	Howard	Raj	Mary	Leslie	Kurt	Gablehauser	Doug	Summer
Leonard	198 91%	14 6%	1 0%	1 0%	1 0%	2 1%	1 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Sheldon	14 6%	224 90%	6 2%	2 1%	1 0%	3 1%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Penny	1 1%	4 4%	90 93%	2 2%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Unknown	1 10%	0 0%	0 0%	9 90%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Howard	3 7%	2 4%	0 0%	0 0%	41 89%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Raj	0 0%	1 4%	0 0%	0 0%	1 4%	25 93%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Mary	1 3%	2 5%	0 0%	0 0%	3 8%	0 0%	33 85%	0 0%	0 0%	0 0%	0 0%	0 0%
Leslie	2 20%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	8 80%	0 0%	0 0%	0 0%	0 0%
Kurt	0 0%	0 0%	2 67%	0 0%	0 0%	0 0%	0 0%	0 0%	1 33%	0 0%	0 0%	0 0%
Gablehauser	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	2 100%	0 0%	0 0%
Doug	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -
Summer	0 0%	0 0%	0 0%	1 100%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%

Figure B.1.: Confusion matrix of the speaker assignment on BBT. Since *Leonard* and *Sheldon* appear most often with other characters on the screen, they are target of the most confusion. Similarly, their face tracks are more often confused with other characters.



	Buffy	Riley	Xander	Willow	Unknown	Giles	Dawn	Anya	Tara	Spike	Harmony	Xander2	Joyce	Glory	Dracula	Maclay	Beth	Graham	Overheiser	Mort	Leiach	Donny	Manager	Ben	Watchman	Sandy	Toth	Monk
Buffy	260 90%	10 3%	3 1%	3 1%	1 0%	2 1%	4 1%	2 1%	0 0%	0 0%	0 0%	0 0%	2 1%	1 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Riley	6 8%	68 88%	1 1%	1 1%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	1 1%	0 0%	0 0%
Xander	8 7%	2 2%	94 77%	4 3%	2 2%	0 0%	1 1%	2 2%	2 2%	0 0%	0 0%	5 4%	0 0%	0 0%	1 1%	0 0%	0 0%	0 0%	0 0%	1 1%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Willow	0 0%	2 2%	3 3%	98 91%	1 1%	1 1%	0 0%	0 0%	3 3%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Unknown	0 0%	0 0%	0 0%	0 0%	12 80%	0 0%	1 7%	0 0%	0 0%	0 0%	1 7%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	1 7%	0 0%	0 0%	0 0%	0 0%
Giles	4 6%	1 2%	1 2%	4 6%	0 0%	52 79%	3 5%	1 2%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Dawn	3 7%	2 5%	0 0%	0 0%	0 0%	1 2%	38 86%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Anya	2 4%	0 0%	1 2%	2 4%	1 2%	1 2%	0 0%	43 81%	0 0%	0 0%	0 0%	2 4%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	1 2%	0 0%	0 0%	0 0%	0 0%	0 0%
Tara	1 3%	1 3%	0 0%	3 9%	0 0%	0 0%	1 3%	1 3%	26 79%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Spike	1 2%	0 0%	0 0%	0 0%	1 2%	0 0%	0 0%	0 0%	0 0%	45 90%	1 2%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	2 4%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Harmony	1 1%	0 0%	0 0%	0 0%	2 3%	0 0%	3 4%	0 0%	0 0%	3 4%	66 88%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Xander2	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	1 7%	0 0%	0 0%	0 0%	14 93%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Joyce	1 3%	0 0%	0 0%	0 0%	0 0%	0 0%	4 13%	0 0%	0 0%	0 0%	0 0%	0 0%	26 84%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Glory	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	27 100%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Dracula	1 20%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	4 80%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Maclay	0 0%	0 0%	0 0%	1 6%	0 0%	1 6%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	13 76%	1 6%	0 0%	0 0%	0 0%	0 0%	1 6%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Beth	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	1 6%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	16 94%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Graham	1 7%	2 13%	0 0%	0 0%	1 7%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	11 73%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Overheiser	1 12%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	1 12%	2 25%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	4 50%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Mort	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	2 50%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	2 50%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Leiach	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -
Donny	0 0%	0 0%	0 0%	0 0%	0 0%	1 12%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	7 88%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
Manager	0 0%	1 17%	1 17%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	4 67%	0 0%	0 0%	0 0%	0 0%	0 0%
Ben	0 0%	0 0%	0 0%	0 0%	1 14%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	6 86%	0 0%	0 0%	0 0%	0 0%
Watchman	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	5 100%	0 0%	0 0%	0 0%
Sandy	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	2 100%	0 0%	0 0%
Toth	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -
Monk	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -

Figure B.2.: Confusion matrix of the speaker assignment on BUFFY. The characters *Buffy* and *Xander* get assigned most tracks from other characters. Also, most of the other main characters are assigned at least one *Buffy* track. Especially for identities with a low absolute number of identity assignments this can be a cause for later confusion with *Buffy* during identification.

Appendix C

Face ID result analysis for BUFFY

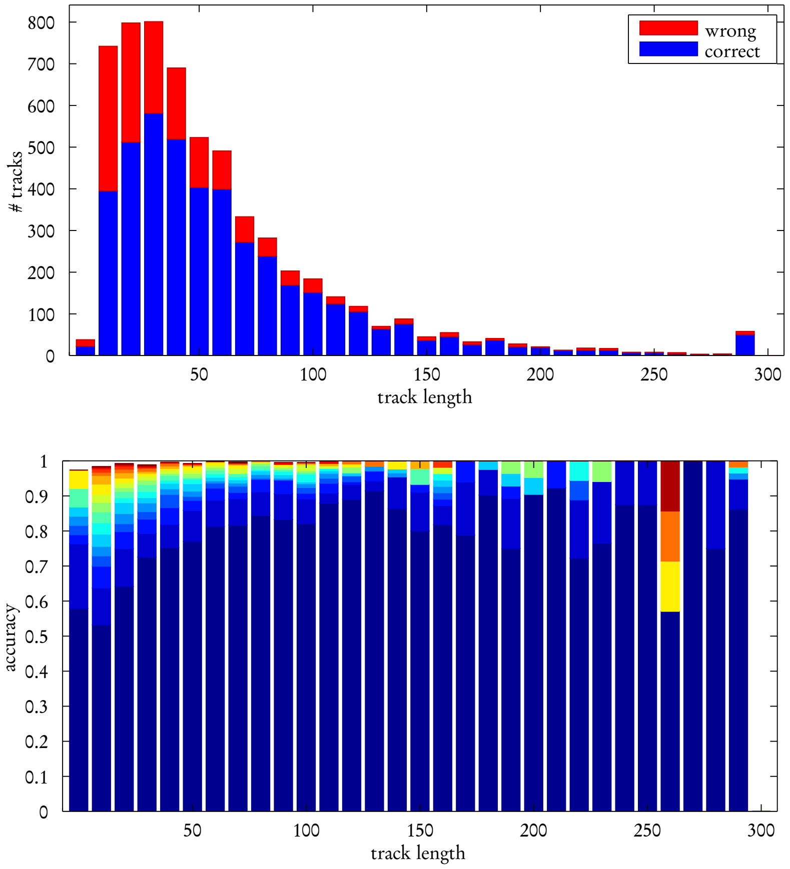


Figure C.1.: Track-level recognition accuracy in dependency of track length for BUFFY. (top) Absolute numbers of correctly/wrongly labeled tracks, (bottom) relative performance over multiple ranks.

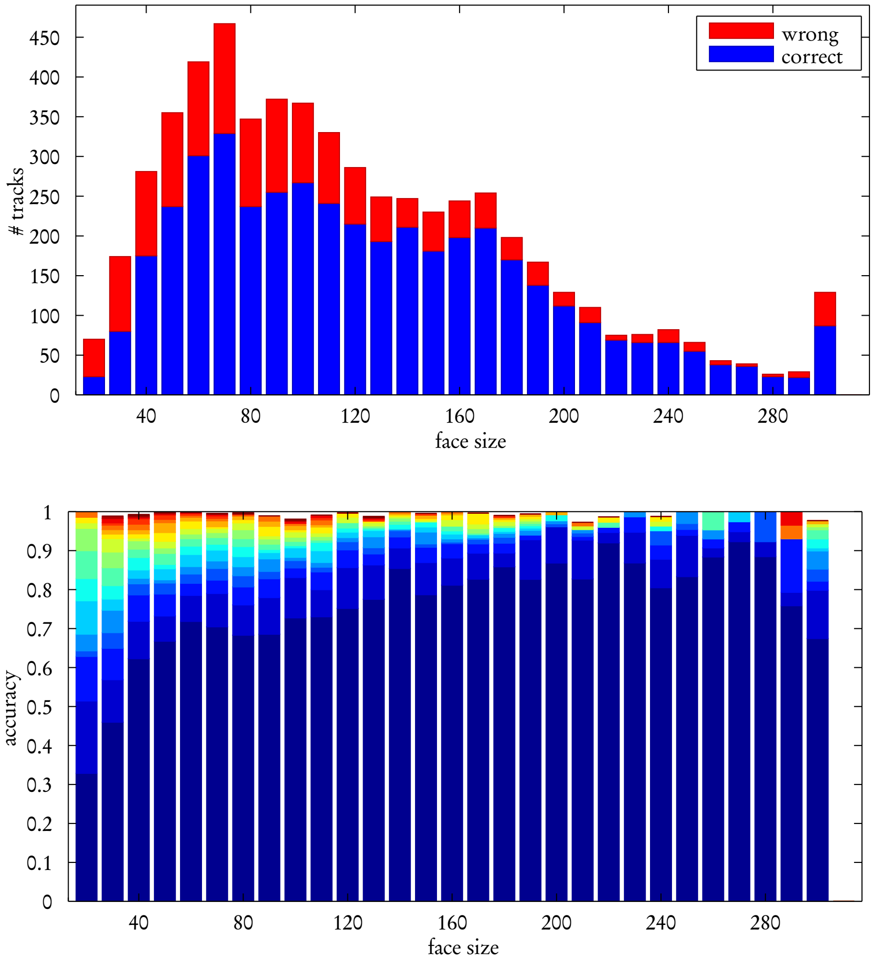


Figure C.2.: Track-level recognition accuracy in dependency of face size for BUFFY. (top) Absolute numbers of correctly/wrongly labeled tracks, (bottom) relative performance over multiple ranks.

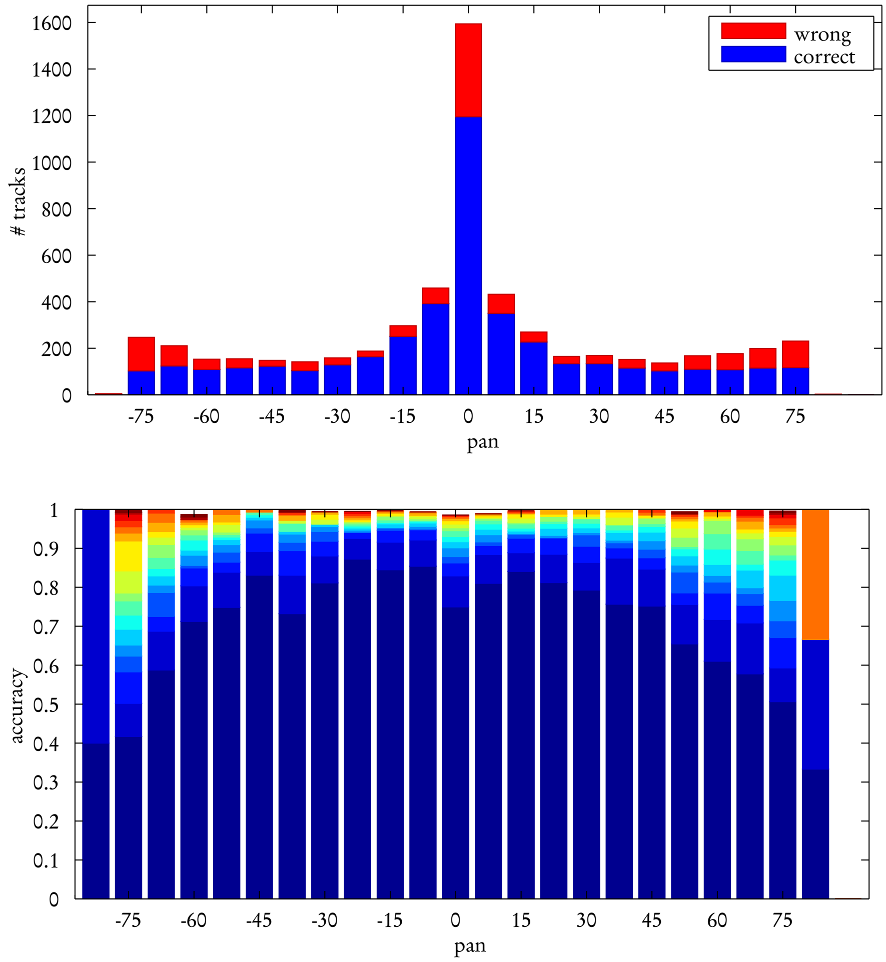


Figure C.3.: Track-level recognition accuracy in dependency of pan angle for BUFFY. (top) Absolute numbers of correctly/wrongly labeled tracks, (bottom) relative performance over multiple ranks.

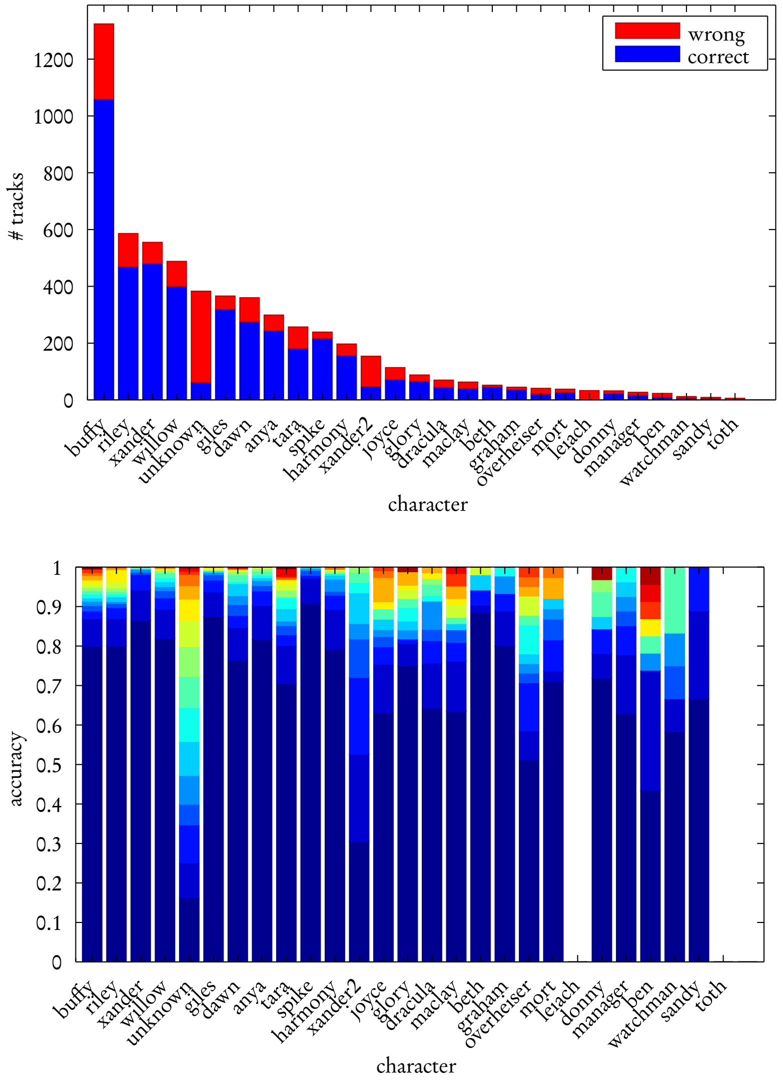


Figure C.4.: Track-level recognition accuracy in dependency of different characters for BUFFY. (top) Absolute numbers of correctly/wrongly labeled tracks, (bottom) relative performance over multiple ranks. *Unknowns* exhibit very poor performance, which is also reflected in the high false acceptance rate. Similarly, characters with low number of tracks (and only few training samples) perform worse than the main characters, for which we obtain around 90% accuracy.

Own Publications

Martin Bäuml, Keni Bernardin, Mika Fischer, Hazim Ekenel, and Rainer Stiefelhagen. Multi-Pose Face Recognition for Person Retrieval in Camera Networks. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2010a. 7, 10, 58

Martin Bäuml, Mika Fischer, Keni Bernardin, Hazim K Ekenel, and Rainer Stiefelhagen. Interactive person-retrieval in TV series and distributed surveillance video. In *ACM Multimedia (demo program)*, 2010b. 3

Martin Bäuml and Rainer Stiefelhagen. Evaluation of local features for person re-identification in image sequences. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011. 75

Michael Weber, Martin Bäuml, and Rainer Stiefelhagen. Part-based clothing segmentation for person retrieval. In *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2011. 5, 75

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. "Knock! Knock! Who is it?" Probabilistic Person Identification in TV-Series. In *CVPR*, 2012. 7

Alexander Schick, Martin Bäuml, and Rainer Stiefelhagen. Improving foreground segmentations with probabilistic superpixel Markov random fields. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2012.

Markus Roth, Martin Bäuml, Ram Nevatia, and Rainer Stiefelhagen. Robust Multi-Pose Face Tracking by Multi-Stage Tracklet Association. In *ICPR*, 2012. 35, 36, 37

Martin Bäuml, Makarand Tapaswi, Arne Schumann, and Rainer Stiefelhagen. Contextual Constraints for Person Retrieval in Camera Networks. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2012. 3

Martin Bäuml, Makarand Tapaswi, and Rainer Stiefelhagen. Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. In *CVPR*, 2013. 7

Arne Schumann, Martin Bäuml, and Rainer Stiefelhagen. Person tracking-by-detection with efficient selection of part-detectors. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2013.

Martin Bäuml, Makarand Tapaswi, and Rainer Stiefelhagen. A Time Pooled Track Kernel for Person Identification. In *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2014. 7

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Story-based Video Retrieval in TV series using Plot Synopses. In *International Conference on Multimedia Retrieval (ICMR)*, 2014a.

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. StoryGraphs : Visualizing Character Interactions as a Timeline. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014b.

Makarand Tapaswi, Cemal Cagri Cörez, Martin Bäuml, Hazim Kemal Ekenel, and Rainer Stiefelhagen. Cleaning up after a Face Tracker: False Positive Removal. In *International Conference on Image Processing (ICIP)*, 2014c. 22, 51

Bibliography

T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision (ECCV)*, 2004. 41

M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 11, 12

D. Anguelov, K.-C. Lee, S. B. Gokturk, and B. Sumengen. Contextual Identity Recognition in Personal Photo Albums. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 5, 42, 44, 75, 76

J. Annesley, J. Orwell, and J. P. Renno. Evaluation of MPEG7 color descriptors for visual surveillance retrieval. In *Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005. 75

O. Arandjelovic and R. Cipolla. Automatic cast listing in feature-length films with anisotropic manifold space. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 4

S. Ba and J. Odobez. A probabilistic framework for joint head tracking and pose estimation. In *International Conference on Patter Recognition (ICPR)*, 2004. 15, 16

B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 11, 12, 13

S. Bak, E. Corvee, F. Bremond, and M. Thonnat. Multiple-shot human re-identification by Mean Riemannian Covariance Grid. In *International Conference on Advanced*

Video and Signal-Based Surveillance (AVSS), 2011. 75

M. Bäuml and R. Stiefelhagen. Evaluation of local features for person re-identification in image sequences. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011. 75

M. Bäuml, K. Bernardin, M. Fischer, H. Ekenel, and R. Stiefelhagen. Multi-Pose Face Recognition for Person Retrieval in Camera Networks. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2010a. 7, 10, 58

M. Bäuml, M. Fischer, K. Bernardin, H. K. Ekenel, and R. Stiefelhagen. Interactive person-retrieval in TV series and distributed surveillance video. In *ACM Multimedia (demo program)*, 2010b. 3

M. Bäuml, M. Tapaswi, A. Schumann, and R. Stiefelhagen. Contextual Constraints for Person Retrieval in Camera Networks. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2012. 3

M. Bäuml, M. Tapaswi, and R. Stiefelhagen. Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 7

M. Bäuml, M. Tapaswi, and R. Stiefelhagen. A Time Pooled Track Kernel for Person Identification. In *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2014. 7

R. Benenson, M. Mathias, R. Timofte, and Van Go. Pedestrian detection at 100 frames per second. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 15

J. Berclaz and F. Fleuret. Multiple object tracking using k-shortest paths optimization. *Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011. 13, 14

T. Berg and P. Belhumeur. Tom-vs-Pete Classifiers and Identity-Preserving Alignment for Face Verification. In *British Machine Vision Conference (BMVC)*, 2012. 41

T. Berg, A. Berg, J. Edwards, M. Maire, R. White, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 3, 41

K. Bernardin and R. Stiefelhagen. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing*, 2008: 1–10, 2008. 23

K. Bernardin, R. Stiefelhagen, and A. Waibel. Probabilistic integration of sparse audio-visual cues for identity tracking. In *ACM International Conference on Multimedia (ACMMM)*, 2008. 58, 76

M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. *International Conference on Machine Learning (ICML)*, 2004. 43

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Computational Learning Theory (COLT)*, 1998. 42

P. Bojanowski, F. Bach, I. Laptev, and J. Ponce. Finding Actors and Actions in Movies. In *International Conference on Computer Vision (ICCV)*, 2013. 41

A. Borràs, F. Tous, J. Llados, and M. Vanrell. High-Level Clothes Description Based on Colour-Texture and Structural Features. *Pattern Recognition and Image Analysis*, pages 108–116, 2003. 75

L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *International Conference on Computer Vision (ICCV)*, 2009. 11, 12, 78, 79

J. E. Boyd and J. J. Little. Biometric Gait Recognition. *Advanced Studies in Biometrics*, pages 19–42, 2005. 74

Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998. 76

Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *International Conference on Computer Vision (ICCV)*, number July, 2001. 76

H. Bredin, J. Poignant, M. Tapaswi, G. Fortier, V. B. Le, T. Napoleon, H. Gao, C. Barras, S. Rosset, L. Besacier, J. Verbeek, G. Quenot, F. Jurie, and H. K. Ekenel. Fusion of speech, faces and text for person identification in TV broadcast. In *ECCV Workshop on Information Fusion in Computer Vision for Concept Recognition*, 2012. 76

M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *International Conference on Computer Vision (ICCV)*, 2009. 11, 13

H. Cevikalp and B. Triggs. Face recognition based on image sets. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 100

O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2004. 42

A. Chapelle, Olivier Schölkopf, Bernhard Zien. A Discussion of Semi-Supervised Learning and Transduction. In *Semi-supervised Learning*, chapter 25. MIT Press Cambridge, MA, USA, 1st edition, 2006. 40

C. Chen, A. Heili, and J.-M. Odobez. Combined Estimation of Location and Body Pose in Surveillance Video. In *Advanced Video and Signal Based Surveillance*, 2011. 15, 16

D. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom Pictorial Structures for Re-identification. In *British Machine Vision Conference (BMVC)*, 2011. 75

R. Cinbis, J. Verbeek, and C. Schmid. Unsupervised metric learning for face identification in TV video. In *International Conference on Computer Vision (ICCV)*, 2011. 5, 43

R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. *Journal of Machine Learning Research*, 7:1687–1712, 2006. 42

T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6, 41, 76

T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *European Conference on Computer Vision (ECCV)*, 2008. 6

T. Cour, B. Sapp, A. Nagle, and B. Taskar. Talking Pictures: Temporal Grouping and Dialog-Supervised Person Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 4, 53

T. Cour, B. Sapp, and B. Taskar. Learning from Partial Labels. *Journal of Machine Learning Research*, 12(5):1225–1261, 2011. 4, 6

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 11, 12, 78, 79

M. Dikmen, E. Akbas, T. Huang, and N. Ahuja. Pedestrian recognition with a learned metric. In *Asian Conference on Computer Vision (ACCV)*, 2010. 75

P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *British Machine Vision Conference (BMVC)*, pages 1–11, 2009. 12

P. Dollár, S. Belongie, and P. Perona. The Fastest Pedestrian Detector in the West. In *British Machine Vision Conference (BMVC)*, 2010. 15

H. K. Ekenel, M. Fischer, Q. Jin, and R. Stiefelhagen. Multi-modal Person Identification in a Smart Environment. In *CVPR Workshop on Biometrics*, 2007a. 76

H. K. Ekenel, J. Stallkamp, H. Gao, M. Fischer, and R. Stiefelhagen. Face Recognition for Smart Interactions. In *International Conference on Multimedia and Expo*, 2007c. 58

H. Ekenel and R. Stiefelhagen. Analysis of Local Appearance-Based Face Recognition: Effects of Feature Selection and Feature Normalization. *Conference on Computer Vision and Pattern Recognition Workshop*, 2006. 41, 56, 58

M. Everingham, J. Sivic, and A. Zisserman. ”Hello! My name is... Buffy” – Automatic naming of characters in TV video. In *British Machine Vision Conference (BMVC)*, 2006. 4, 5, 14, 39, 41, 44, 51, 52, 53, 60, 63, 74, 75, 76

M. Farenzena, L. Bazzani, A. Perina, and V. Person Re-Identification by Symmetry-Driven Accumulation of Local Features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 74, 75

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2009. 11, 12, 78

M. Fischer, H. K. Ekenel, and R. Stiefelhagen. Person re-identification in TV series using robust face recognition and user feedback. *Multimedia Tools and Applications*, 2010. 4, 5, 58

M. Fischer, H. K. Ekenel, and R. Stiefelhagen. Analysis of partial least squares for pose-invariant face recognition. In *International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2012. 41

A. Fitzgibbon and A. Zisserman. Joint manifold distance: a new approach to appearance based clustering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 4

Y. Freund and R. E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1): 119–139, 1997. 12

B. Fröba and A. Ernst. Face detection with the modified census transform. In *Automatic Face and Gesture Recognition (FG)*, 2004. 11, 12

A. C. Gallagher and T. Chen. Clothing cosegmentation for recognizing people. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008a. 4, 5, 43, 73, 75, 76

A. Gallagher and T. Chen. Using a markov network to recognize people in consumer images. In *International Conference on Image Processing (ICIP)*, 2007. 4, 42, 76

A. Gallagher and T. Chen. Estimating age, gender, and identity using first name priors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008b. 76

A. Gallagher and T. Chen. Understanding images of groups of people. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 76

D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *International Conference on Computer Vision (ICCV)*, 1999. 14

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Transactions on Pattern Analysis and Machine Intelligence*, 6(6): 721–741, 1984. 76

N. Gheissari, T. B. Sebastian, and R. Hartley. Person Reidentification Using Spatiotemporal Appearance. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2, 2006. 74

F. Gianni, J. Pinquier, and E. Irisa. Acadi showcase - automatic character indexing in audiovisual document. In *Conference on Image and Video Retrieval (CIVR)*, 2007. 5

J. Good. How many photos have ever been taken?, 2011. URL `http://blog.1000memories.com/94-number-of-photos-ever-taken-digital-and-analog-in-shoebox`. 1

N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140:107–113, 1993. 18

Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Conference on Neural Information Processing Systems (NIPS)*, 2005. 42, 44, 46

D. Gray and H. Tao. Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. In *European Conference on Computer Vision (ECCV)*, 2008. 74, 75

D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, 51(2):271–279, 1989. 76

M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *European Conference on Computer Vision (ECCV)*, 2010. 3

M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Face Recognition from Caption-Based Supervision. *International Journal of Computer Vision*, 96(1):64–82, 2012. 3, 43

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2009. 47

B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machines: Global versus component-based approach. In *International Conference on Computer Vision (ICCV)*, 2001. 41

M. Hirzer, P. Roth, M. Köstinger, and H. Bischof. Relaxed Pairwise Learned Metric for Person Re-identification. *European Conference on Computer Vision (ECCV)*, 2012. 76

M. Hofmann, D. Wolf, and G. Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 13, 14

S. Hoi, W. Liu, M. Lyu, and W. Ma. Learning distance metrics with contextual constraints for image retrieval. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 43

Y. Hu, A. S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 99, 100

C. Huang, B. Wu, and R. Nevatia. Robust Object Tracking by Hierarchical Association of Detection Responses. In *European Conference on Computer Vision (ECCV)*, 2008. 13, 14, 35, 79

M. Hunke and A. Waibel. Face locating and tracking for human-computer interaction. In *Asilomar Conference on Signals, Systems and Computers*, 1994. 11

M. Isard and A. Blake. CONDENSATION - Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998a. 18

M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *International Conference on Computer Vision (ICCV)*, 1998b. 13

ISO/IEC 15938-3. Multimedia Content Description Interface - Part 3: Visual, 2001. 80

G. Jaffré and P. Joly. Costume: A New Feature for Automatic Video Content Indexing. In *Proceedings Recherche d'Information Assiste par Ordinateur*, pages 314–325, 2004. 75

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

H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 13, 14

T. Joachims. Transductive inference for text classification using support vector machines. *International Conference on Machine Learning (ICML)*, 1999. 42

A. Joulin and F. Bach. A convex relaxation for weakly supervised classifiers. In *International Conference on Machine Learning (ICML)*, 2012. 42

Z. Kalal, J. Matas, and K. Mikolajczyk. PN learning: Bootstrapping binary classifiers by structural constraints. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010a. 11, 12

R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(Series D):35–45, 1960. 18

S. Khan and M. Shah. Tracking multiple occluding people by localizing on multiple scene planes. *Transactions on Pattern Analysis and Machine Intelligence*, 31(3):505–519, 2008. 13

E. Khoury, P. Gay, and J. Odobez. Fusing matching and biometric similarity measures for face diarization in video. *International Conference on Multimedia Retrieval (ICMR)*, 2013. 3, 5

M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 15

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 1971. 48

J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998. 59, 76

R. Kondor and T. Jebara. A kernel between sets of vectors. *International Conference on Machine Learning (ICML)*, 2003. 102

M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof. Learning to Recognize Faces from Videos and Weakly Related Information Cues. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011. 4, 42, 49, 51

M. Köstinger, M. Hirzer, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 76

N. Krahnstoever, M.-C. Chang, and W. Ge. Gaze and Body Pose Estimation from a Distance. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011. 15, 16

C. Küblbeck and A. Ernst. Face detection and tracking in video sequences using the modified census transformation. *Image and Vision Computing*, 24(6):564–572, 2006. 11, 12, 14, 21, 22, 28

N. Kumar and A. Berg. Attribute and simile classifiers for face verification. In *International Conference on Computer Vision (ICCV)*, 2009. 73

C.-h. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 12

O. Lanz. Approximate Bayesian multibody tracking. *Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1436–49, 2006. 19

R. Layne, T. Hospedales, and S. Gong. Towards person identification and re-identification with attributes. In *European Conference on Computer Vision (ECCV)*, 2012. 75

Y. Lee and O. Mangasarian. RSVM: Reduced Support Vector Machines. In *SIAM International Conference on Data Mining*, 2001. 51, 62

B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1):259–289, 2008. 11, 12

B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *International Conference on Computer Vision (ICCV)*, 2007. 13

V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet physics doklady*, 1966. 53

D. Li, G. Wei, I. K. Sethi, and N. Dimitrova. Fusion of visual and audio features for person identification in real video. In *Proc. SPIE 4315, Storage and Retrieval for Media Databases*, volume 4315, 2001. 5

W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 76

Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 13, 14

Z. Li and J. Liu. Constrained clustering by spectral kernel learning. *International Conference on Computer Vision (ICCV)*, 2009. 43

D. Lin, A. Kapoor, and G. Hua. Joint people, event, and location recognition in personal photo collections using cross-domain context. *Artificial Intelligence*, 2010. 4, 5, 42, 44, 76, 77

D. C. Liu and J. Nocedal. On the limited memory BGFS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989. 51

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002. 102

D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 79

S. Lyu. Mercer Kernels for Object Recognition with Local Features. Technical Report October, 2004. 103

S. Lyu. Mercer Kernels for Object Recognition with Local Features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 100, 101, 103, 105

S. Maji and J. Malik. Object detection using a max-margin Hough transform. *Conference on Computer Vision and Pattern Recognition*, 2009. 12

S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 100

K. V. Mardia and P. Jupp. *Directional Statistics*. John Wiley and Sons Ltd., 2nd editio edition, 1999. 20

S. Melacci, M. Maggini, and M. Gori. Semi-supervised learning with constraints for multiview object recognition. In *International Conference on Artificial Neural Networks (ICANN)*, 2009. 43

R. Morzinger, M. Thaler, S. Stalder, H. Grabner, and L. Van Gool. Improved person detection in industrial environments using multiple self-calibrated cameras. *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011. 16

C. Nakajima, M. Pontil, B. Heisele, and T. Poggio. Full-body person recognition system. *Pattern Recognition*, 36:1997–2006, 2003. 74

M. Nechyba, L. Brandy, and H. Schneiderman. Pittpatt face detection and tracking for the CLEAR 2007 evaluation. In *Multimodal Technologies for Perception of Humans*, pages 126–137. Springer, 2008. 14

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine learning*, 39(2-3):103–134, 2000. 42

J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006. 87

E. G. Ortiz, A. Wright, and M. Shah. Face Recognition in Movie Trailers via Mean Sequence Sparse Representation-Based Classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 4, 5, 41

O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A Compact and Discriminative Face Track Descriptor. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5, 62, 63, 100

J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. In *Microsoft Research TR-98-14*, 1998. 102

J. Poignant, H. Bredin, and V. Le. Unsupervised speaker identification using overlaid texts in TV broadcast. In *InterSpeech*, 2012. 3

B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang. Person Re-Identification by Support Vector Ranking. In *British Machine Vision Conference (BMVC)*, 2010. 75

D. Ramanan, S. Baker, and S. Kakade. Leveraging archival video for building face datasets. In *International Conference on Computer Vision (ICCV)*, 2007a. 4, 5, 75, 76

D. Ramanan, D. a. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 29(1): 65–81, 2007b. 5, 12, 43

K. Rematas, M. Fritz, and T. Tuytelaars. The pooled NBNN kernel: beyond image-to-class and image-to-image. In *Asian Conference on Computer Vision (ACCV)*, 2012. 100, 103, 104

D. Rim, K. Hassan, and C. Pal. Semi Supervised Learning for Wild Faces and Video. In *British Machine Vision Conference (BMVC)*, 2011. 42

M. Roth, M. Bäuml, R. Nevatia, and R. Stiefelhagen. Robust Multi-Pose Face Tracking by Multi-Stage Tracklet Association. In *International Conference on Pattern Recognition (ICPR)*, 2012. 35, 36, 37

C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 76

D. Salvi, J. Waggoner, A. Temlyakov, and S. Wang. A Graph-Based Algorithm for Multi-Target Tracking with Occlusion. 2012. 14

J. Sang and C. Xu. Robust face-name graph matching for movie character identification. *Transactions on Multimedia*, 14(3):586–596, 2012. 6

P. Sankar, C. Jawahar, and A. Zisserman. Subtitle-free Movie to Script Alignment. In *British Machine Vision Conference (BMVC)*, 2009. 4

M. Sargin and H. Aradhye. Audiovisual celebrity recognition in unconstrained web videos. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009. 4, 5

W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human Detection Using Partial Least Squares Analysis. *International Conference on Computer Vision (ICCV)*, 2009. 12

E. Seemann, M. Fritz, and B. Schiele. Towards robust pedestrian detection in crowded image sequences. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 12

B. Settles. Active learning literature survey. Technical report, 2010. URL `http://csis.bits-pilani.ac.in/faculty/goel/course_material/MachineLearning/2013/ReadingMaterial/settles.activelearning.pdf`. 108

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient sOLver for SVM. In *International Conference on Machine Learning (ICML)*, 2007. 100

P. Sharma, C. Huang, and R. Nevatia. Unsupervised incremental learning for improved object detection in a video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 13

J. Shi, R. Liao, and J. Jia. CoDeL: A Human Co-detection and Labeling Framework. In *International Conference on Computer Vision (ICCV)*, 2013. 75

J. Sivic, C. L. Zitnick, and R. Szeliski. Finding people in repeated shots of the same scene. In *British Machine Vision Conference (BMVC)*, 2006a. 75

J. Sivic, M. Everingham, and A. Zisserman. “Who are you?” – Learning person specific classifiers from video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 4, 5, 14, 35, 36, 37, 41, 51, 55, 60, 62, 63

J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *International Conference on Image and Video Retrieval*, 2005. 4, 5

X. Song, C. Lin, and M. Sun. Cross-modality automatic face model training from large video databases. In *CVPR Workshop*, 2004. 3

J. Stallkamp, H. K. Ekenel, and R. Stiefelhagen. Video-based Face Recognition on Real-World Data. *International Conference on Computer Vision (ICCV)*, 2007. 58

R. Stiefelhagen. Tracking focus of attention in meetings. *International Conference on Multimodal Interaction (ICMI)*, 2002. 15

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 41

M. Tapaswi, M. Bäuml, and R. Stiefelhagen. "Knock! Knock! Who is it?" Probabilistic Person Identification in TV-Series. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 7

M. Tapaswi, C. C. Cörez, M. Bäuml, H. K. Ekenel, and R. Stiefelhagen. Cleaning up after a Face Tracker: False Positive Removal. In *International Conference on Image Processing (ICIP)*, 2014c. 22, 51

M. M. Tapaswi. A Global Model for Person Identification in TV Series. Master's thesis, 2011. 7, 78

D. Vaquero, R. S. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk. Attribute-based people search in surveillance environments. In *Workshop on Applications of Computer Vision (WACV)*, 2009. 75

A. Vedaldi and A. Zisserman. Sparse kernel approximations for efficient classification and detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 100

R. Vezzani, D. Baltieri, and R. Cucchiara. People reidentification in surveillance and forensics. *ACM Computing Surveys*, 46(2):1–37, 2013. 74

P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 12, 14

M. Voit and R. Stiefelhagen. Multi-View Head Pose Estimation using Neural Networks. *Canadian Conference on Computer and Robot Vision*, 2005. 15

M. Voit and R. Stiefelhagen. Tracking head pose and focus of attention with multiple far-field cameras. In *International Conference on Multimodal Interaction (ICMI)*, 2006. 15

C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *International Conference on Computer Vision (ICCV)*, 2003. 100

R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-Manifold Distance with application to face recognition based on image set. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 100

R. Wang, H. Guo, L. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 100, 109, 110

J. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. 81

M. Weber, M. Bäuml, and R. Stiefelhagen. Part-based clothing segmentation for person retrieval. In *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2011. 5, 75

C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *Transactions on Pattern Analysis and Machine Intelligence*, 19: 780–785, 1997. 11

B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007. 11, 12, 13, 14

B. Wu, Y. Zhang, B.-G. Hu, and Q. Ji. Constrained Clustering and Its Application to Face Clustering in Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013b. 43

B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 13

B. Wu and R. Nevatia. Detection and Segmentation of Multiple, Partially Occluded Objects by Grouping, Merging, Assigning Part Detection Responses. *International Journal of Computer Vision*, 82(2):185–204, 2009. 12

X. Xiong and F. D. la Torre. Supervised descent method and its applications to face alignment. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 53

L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *Conference on Neural Information Processing Systems (NIPS)*, 2005. 42

K. Yamaguchi, M. Kiapour, L. Ortiz, and T. Berg. Parsing clothing in fashion photographs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5, 75

K. Yamamoto, O. Yamaguchi, and H. Aoki. Fast face clustering based on shot similarity for browsing video. *Progress in Informatics*, (7):53–62, 2010. 4, 6

R. Yan, J. Zhang, J. Yang, and A. G. Hauptmann. A discriminative learning framework with pairwise constraints for video object classification. *Transactions on Pattern Analysis and Machine Intelligence*, 28(4):578–93, 2006. 43, 46, 58

B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012a. 14

B. Yang and R. Nevatia. Online Learned Discriminative Part-Based Appearance Models for Multi-Human Tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012b. 13, 14

J. Yang and A. Waibel. A real-time face tracker. In *Workshop on Applications of Computer Vision (WACV)*, 1996. 11, 14

J. Yang, R. Yan, and A. G. Hauptmann. Multiple instance learning for labeling faces in broadcasting news video. In *ACM International Conference on Multimedia (ACMMM)*, 2005. 3

T. Yu, Y. Yao, D. Gao, and P. Tu. Learning to Recognize People in a Smart Environment. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011. 43

Y. Yusoff, W. Christmas, and J. Kittler. A Study on Automatic Shot Change Detection. *Multimedia Applications and Services*, 1998. 27

B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. On-line Semi-supervised Multiple-Instance Boosting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 42, 49

H. Zeng and Y.-m. Cheung. Semi-supervised maximum margin clustering with pairwise constraints. *Transactions on Knowledge and Data Engineering*, 24(5):926–939, 2012. 43

G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu. Boosting Local Binary Pattern (LBP)-Based Face Recognition. *Advances In Biometric Person Authentication*, 3338 (LNCS):179–186, 2004. 41

L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 13, 14

L. Zhang, D. Kalashnikov, and S. Mehrotra. A Unified Framework for Context Assisted Face Clustering. In *International Conference on Multimedia Retrieval (ICMR)*, 2013. 4, 5

W. Zhang, T. Zhang, and D. Tretter. Clothing-based Person Clustering in Family Photos. In *International Conference on Image Processing (ICIP)*, 2010. 75

Y. Zhang, C. Xu, H. Lu, and Y. Huang. Character identification in feature-length films using global face-name matching. *Transactions on Multimedia*, 11(7):1276–1288, 2009. 6

M. Zhao and J. Yagnik. Large scale learning and recognition of faces in web videos. *Automatic Face and Gesture Recognition (FG)*, 2008. 4

W. Zhao and R. Chellappa. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003. 41

X. Zhao, N. Evans, and J. Dugelay. Semi-supervised face recognition with LDA self-training. In *International Conference on Image Processing (ICIP)*, 2011. 42