Monitoring Interactions

Felix Meißner

ICA-3623424

Supervisor: Remco Veltkamp, Robby T. Tan, Ronald Poppe

Universiteit Utrecht
March 22, 2016

Abstract

This work proposes a human interaction recognition based approach to video indexing that represents a video by showing when and with whom was interacted throughout the video. In order to visualize the length of an interaction, it is required to recognize individuals that have been detected in earlier parts of the video. To solve this problem, an approach to photo-clustering is extended to video material by tracking detected faces and using the information from tracking to improve the recognition of human beings. The results of the tracking based approach show a considerable decrease of false cluster assignments compared to the original method. Further, it is demonstrated that the proposed method is able to correctly recognize the appearance of five out of the six individuals correctly.

Contents

1 Introduction

1.1 Project description

With the recent development in digital video camera technology, it becomes feasible for people to record their daily lives from an egocentric point of view. For example, the company Looxcie currently offers a head mountable camera which can record up to 4 hours of 480p video data at a price of no more than 100$. Another supplier of head mounted cameras is GoPro. While their products focus on higher quality but less battery life, it is just a matter of time until more of these or similar devices will provide enough battery power to record a whole day's activities. The probably most spectacular product of this category are smart glasses like Google Glass. Next to being a consumer product, body worn cameras are also used in the public sector, especially by the police like for example in the United Kingdom. Altogether one can say that a growth of egocentric videos can be expected in the near future.

Opposed to conventional recording devices like hand-held cameras, body worn cameras are not used to record specific actions, plots or events. Instead, they are commonly used to record a whole mission, trip or even all day long. The recordings are typically archived in order to be referred to later on, for example by a consumer who wants to remember the experience of a particular day or by a police officer who wants to analyze a particular mission. In order to extract the meaningful parts of the videos later on, a possibility is needed to structure or summarize videos efficiently. Without such techniques, searching for particular moments or events can become a difficult and long lasting task. Many different approaches to this problem can be imagined, and many ideas have already been put into practice. This work specifically focuses on detecting and summarizing the social interactions observed in the video material, in order to aid users that search for particular interactions with other people.

In the remainder of this section, first the detailed goals of this project are described. Second, the approach that has been taken is discussed, as well as related work and its shortcomings that this project aims to overcome. Last, the structure

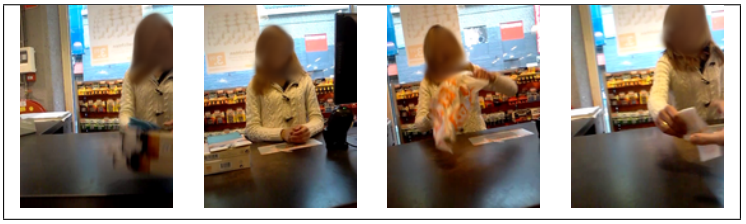


Figure 1.1: An example of an interaction is the interaction of the movie maker with a cashier when paying groceries at a shop.

of the remaining text is given.

1.2 Project goals

The goal of this project is to automatically generate a visual index from egocentric video material. This index should give an overview of the people that interacted with the filmmaker. By interaction any interaction between the filmmaker and a person that is visible in the video is meant, which includes social interactions like meeting friends as well as non-social interactions like paying the groceries at the store. In figure 1.1, several frames are presented that show the cashier of a store interacting with the filmmaker who is paying groceries.

Particularly, only interactions between the filmmaker and other individuals are to be indexed, not interactions between other people in the scene. This restriction is made because the index should help summarizing film-maker's interactions, opposed to recognizing interactions between unknown people.

In figure 1.2 the exemplary output is drafted. The output should show a photo that is extracted from the video, together with a bar that indicates when the interaction started and how long it lasted.

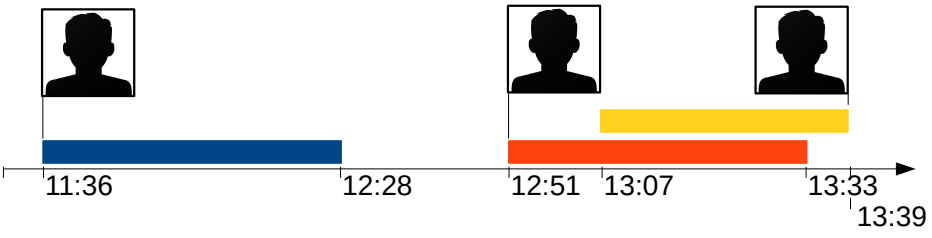


Figure 1.2: Example of an index

It should be possible to automatically generate an index given the recordings of a whole day. The index visually summarizes to which people one has talked to on a given day and thus gives a overview over video material that can be browsed at a glance. With a visual index it becomes possible to search through archived videos efficiently. For example, one can locate the recording of a longer lasting conversation one had with a certain person without the need of watching hours of videos, but by looking for an index showing that person for the expected length of time.

1.3 Approach

Analyzing self recorded egocentric videos revealed that social interactions, for example having lunch with someone, talking to each other or paying groceries at the supermarket, often coincide with looking at each others faces. Vice versa, in egocentric videos, when a particular face is observed in multiple consecutive frames, usually some interaction with that person actually occurred. The detection of frontal faces will therefore be used as an indicator for social interactions with the person belonging to the detected face.

Next to the detection, the system is required to recognize the first and the last frame in which a particular person is detected. This requires the system to recognize if any other person that is detected in a different frame is the same individual as detected before or not.

This identification is approached in two different ways: First, each detected face is visually tracked as long as it remains visible in consecutive frames. Tracking is a reliable technique to answer the question whether a detected face belongs to the same person as a detection from preceding frames. Nonetheless, face tracking may be interrupted before an interaction ends. Such interruptions can for example occur due to head movements of the filmmaker, leading to the interaction partner being out of scene for a moment.

To solve to the problem of identifying individuals in consecutive frames in cases where face tracking had been interrupted, another approach is taken: The detected individuals are visually compared to each other by using a combination of face recognition and appearance based recognition. Later, all detections are clustered according to their pairwise distances. In the optimal case, each cluster contains all detections of a distinct individual.

1.4 Assumptions

The detection of social interactions is based on face detection. It is assumed that the interaction partner is facing towards the camera, which means looking at the face of the filmmaker. Further more it is assumed, that faces of interacting people will be recorded more frequent than faces from non-interacting people, so that it is possible to distinguish between interacting and non-interacting persons by using a threshold on the number of consecutive detections.

As a consequence, another assumption is that faces of interacting people are detected successfully by the algorithm to be developed. This will work for the majority of faces but might fail for people wearing hats, glasses or a scarf that covers parts of the face. The improvement of face detection algorithms is not part of this work and is therefor left open as further research.

1.5 Structure

The remainder of this document is structured as follows: In the following chapter, related research is presented, as well as the distinction to this work. In chapter 3, background information about the employed methods and technologies is given. The detailed approach of this work is explained in chapter 4, followed by its experimental evaluation being presented in chapter 5. The text finishes by discussing the results and pointing out possible directions of further research in chapter 6.

2 Related work

In the following, related work regarding video indexation is presented as well as the relation to the proposed approach of this work. First, different approaches based on key-frame selection are shown. Second, people recognition based approaches are presented.

2.1 Key-frame selection based indexing

A common approach to summarize egocentric video material is to select a representative subset of frames from the complete video. The result of such an approach is a list of thumbnail images that summarizes the video material. Frames are selected in such manner that the resulting subset contains the most important information with the least possible redundancy.

Approaches to video summarization by key-frame selection differ mainly in the technique used for selecting the frames. In the following, three approaches are presented that select key-frames based on image and video features like colors and motion.

In traditional entertainment movies, the scene is a good indicator if frames are related to each other or not. A widely used technique in video summarization is to detect shot boundaries and selecting one or more frames from the resulting subsets.

Scene based key-frame selection is usually accomplished by looking at the differences in color histograms of the frames. Sudden changes in the color histograms are interpreted as a change of scene, and from each scene one or more representing frames are selected. In [2] and [3] for example, scenes are grouped by applying Delaunay clustering to color and texture information, where the cluster centers are used as key-frames that form the summary.

Another approach to key-frame selection is proposed in [4]: First, a similarity graph for all frames is constructed. For this graph a minimum spanning tree

is build, from which edges are removed until a stop-criterion is reached, which depends on the desired number of clusters, as well as the similarity between single frames. Compared to the former approach this allows the user to control the size of the summary.

Image features can help to identify changes in scene, but are difficult to apply to egocentric video material, because scene changes are less abrupt and less common. Therefore, in [5], a sub-shot selection approach tailored for egocentric video material is proposed: Instead of using histograms of colors, the movement patterns are detected and classified into the three categories *static*, *transit* and *head movement*. These categories are recognized by a support vector machine from dense optical flow and blurriness based feature descriptors. Furthermore, [5] proposes a new approach to key-frame selection in egocentric video, which also considers the semantic content. By detecting meaningful objects based on their position and size in the frames as proposed in [6], [5] detects unique objects and their relation to each other by looking at their co-occurrence in a sub-shot. For each detected object one frame is selected, which leads to a object driven video summary including all objects that are important in a sense as in [6].

While the methods described until here can successfully select subsets of frames that represent different scenes or objects, they fail to present the different people shown in the video. Based on the key-frame selection approach, the following methods are focusing on human beings in video material:

In [7], assuming a static camera position, the number and the position of faces is used to build clusters of similar scenes and select key-frames from each cluster. The reasoning behind this is, that the scene is supposed to be unchanged when the number and position of detected faces remains the same. Similar to this, [8] uses features like the number of faces, their size and their position in a spectral clustering approach, in order to select meaningful key-frames from entertainment videos. As both approaches depend on static view angles, they are not applicable to egocentric videos: Important people are often near the frame center, because they are looked at during a conversation. This makes the position and the number of detected faces a unsuitable attribute for detecting unique scenes and persons.

Similar to this, there are approaches that also utilize techniques of face recognition, in order to improve the clustering results: In [9] face detection is applied in order to provide key-frames per person. For face recognition context information is used, like the position of the face detected in a sequence of frames, as well as face recognition methods as described in [10]. Together with the positional

information of face detection, face recognition is used to find important people in a movie by counting their occurrences. As the former approach this method is not expected to perform well on egocentric video material due to the lack of a static camera position.

The survey [11] gives a recent and detailed summary of video summarization techniques similar to the techniques described until here, as well as other summarization approaches that are not considered in this project. However, no other face detection based methods are mentioned. In the following, approaches of people based video indexing are listed.

In general, the key-frame selection approach has the general shortcoming of not being able to construct a concrete index. Instead, a sub-set of all frames is shown. This requires the user to search through enormous amounts of image material, in order to find certain periods of interactions with specified persons. Furthermore, the length of any interaction can only be concluded by finding the first and the last frame showing the person of interest. Compared to this approach, the index that this project proposes should be able to extract and visualize information about the interaction partner, as well as the length of the interaction.

2.2 Human being detection based indexing

Opposed to key-frame selection based approaches, this section presents studies that aimed to also extract and present information about human beings detected in video material. Instead of just selecting frames, here information is actively transformed for the purpose of summarization.

For example, the indexing method is given in [15] creates a video index based on scene change detection using the MPEG-7 visual descriptors. This work also includes an actor time-line, detecting faces using Viola-Jones, and identifying actors using face recognition. Faces are normalized using an Active Shape Model before being matched against a pre-trained face database including known actors. Since the approach can only recognize faces that are known from a training set, it is not applicable to indexing unknown people from egocentric recordings.

In the following, three different approaches to detecting distinct people by applying clustering to detected faces are presented. Compared to the former approach they do not require the people to be known afore.

In [12], "[...] frames of a video sequence are scanned for faces by a Neural Network-based face detector. The extracted faces are then grouped into clusters by a combination of a face recognition method using pseudo two-dimensional Hidden Markov Models and a k-means clustering algorithm" ([12]). The resulting clusters are supposed to represent the different individuals shown in the video.

Conceptual similar to this, [13] summarizes videos based on people shown in the frames. Faces are detected using Viola-Jones and tracked by a particle filter using histograms of oriented gradients (HOG). From the resulting sequences those faces are selected that are useful for face recognition. After calculating a pair-wise distance matrix, faces are grouped by applying constrained agglomerative clustering.

Another approach to clustering faces based on face recognition using two dimensional principal component analysis is made in [14]. Based on face recognition, a pair-wise distance matrix for all detected faces is constructed and clustered by spectral clustering. The results however show that current face recognition alone cannot provide precise information: For a full length video, only 25% of the labeled faces were recognized correctly.

This work approaches the problem of detecting distinct people in a very similar way, namely by spectral clustering. Opposed the former methods, this work aims to extend the face recognition based approach by including context based information to improve the clustering results. Earlier, [16] has used a similarity measure based on clothes, but without including any information from face recognition. As [1] shows, the performance of clustering family pictures can be improved by combining such different information sources, motivating this project to research if such a combination can also improve person based video indexing approaches.

3 Background knowledge

3.1 Face detection

For face detection the OpenCV implementation of the well known method proposed by Paul Viola and Michael Jones is used ([17]).

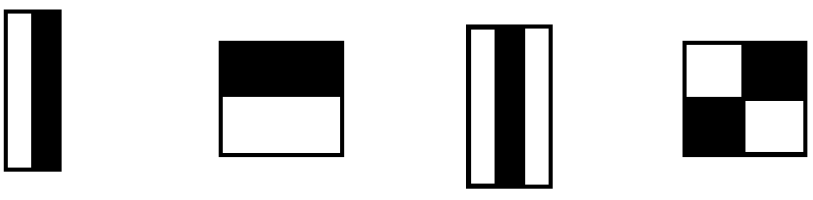
3.1.1 Cascade classifier

The method uses three different haar features, which are shown in figure 3.1:

1. Horizontal and vertical two-rectangle feature (figure 3.1a and figure 3.1b)
2. Three-rectangle feature (figure 3.1c)
3. Four-rectangle feature (figure 3.1d)

Before calculating the features, the image is transformed into a gray scale image, since haar features do not take colors into account. Every feature is then calculated by subtracting the sum of all pixels in the white rectangles from the sum of all pixels in the black rectangles.

These features are calculated at any possible position and scale in a 24 by 24 pixel rectangle in order to build a detector of that size. This will result in more



(a) Horizontal two-rectangle feature (b) Vertical two-rectangle feature (c) Three-rectangle feature (d) Four-rectangle feature

Figure 3.1: Haar features used by Viola-Jones

than 180.000 different values, and using all of them for classification would result in very low efficiency. To overcome this limitation during classification, a small subset of features is selected during the training phase. Given an annotated (face yes or no) training set with images of size 24 by 24 pixels, from all possible haar features that one will be selected, which separates most negative images. The feature type, location and size, as well as the threshold are saved and will be used in the resulting classifier to reject negatives as early as possible.

The resulting classifier is called a cascade classifier, because an image has to pass a sequence of so called weak classifiers in order to be classified as a face. The term weak classifier originates from boosting. In machine learning, boosting means selecting a set of weak classifiers that have high error rates when used isolated, but when combined result in a string classifier with good precision.

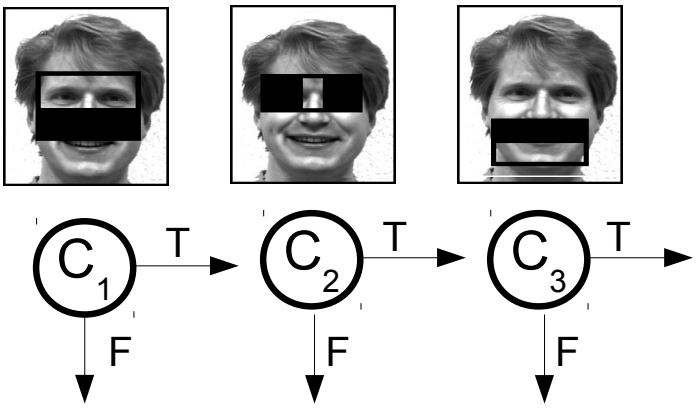


Figure 3.2: Cascade of weak classifiers

Given a 24 by 24 pixel image and the trained classifier, the decision if the image represents a face or not is made as follows: The first haar feature that has been selected in the training phase is applied to the candidate image. If the resulting value is higher than the threshold that has been selected during training, the next feature will be tested. This process is repeated until either one feature is below the threshold given by the classifier, or until there are no features left. When a value that results from a haar feature being applied to the candidate image is below the threshold, the image is immediately classified as a negative, not showing a face. If all resulting values are above the required thresholds, the image is classified as a face. figure 3.2 visualizes this process.

3.1.2 Sliding window

With the classifier described so far it is only possible to classify images of constant size. In order to use the classifier to detect faces in images of variable size a sliding window is used. A rectangular *window* is moved along the image and possibly overlapping patches are extracted. Every patch is resized to the size of the classifier and then tested by it. Every time the rectangle used for extracting those patches reaches the end of the image, its size is increased and the rectangle placed back in the beginning. This process is repeated until the window size cannot be increased anymore, for example because it already has the same size as the whole image, or because it exceeds a given maximum size for expected faces. In figure 3.3 this process is visualized. The patches will not detect any face until their size has been increased enough after several runs.

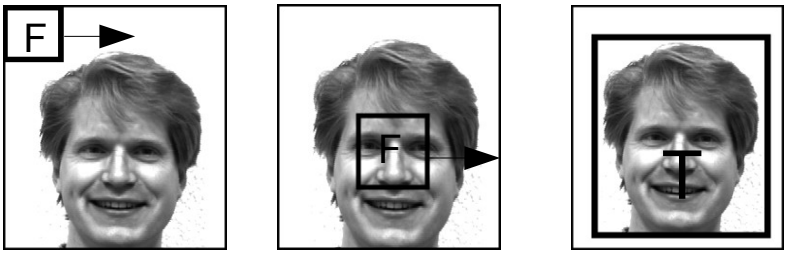


Figure 3.3: Sliding window sequentially tests all possible configurations

Usually, a single face can lead to more than one patch being identified as a face because the sliding window is moved in overlapping steps. In order to provide unambiguous results, OpenCV's implementation of this algorithm will combine detected faces that have large overlaps and return a single detection. By requiring a certain number of such overlapping detections, one can also filter out false alarms.

3.2 Face tracking

For face tracking, three different approaches were tested: Mean shift, particle filter and optical flow.

First the mean shift algorithm implemented in OpenCV has been applied to the video data. The mean shift algorithm works with a color based appearance model which position is updated according to the most likely matching appearance found

in the next frame. In practice difficulties were experienced in detecting whether a face has disappeared or not: When the tracker loses the face, the change of appearance is often not distinguishable from the changes in appearance due to movement and changes of light. Furthermore, if the background has a color model that is similar to model of the face, the tracking is not accurate and returns wrong locations. For this reasons, mean shift has not been used in the final implementation presented in this work.

Another approach to tracking faces are particle filters. Opposed to mean shift, a particle filter modifies the search range based on the confidence that the tracked object is represented by the current position. For each new frame, so called particles are sampled around each original particle from the preceding frame and each particle evaluates the confidence at its position. Particle with high confidence will sample more particles at random positions in short distances to its own location while particles with low confidence will sample less particles at greater distances. The particle with the highest confidence represents the position of the object to be tracked. For the evaluation of the confidence a color based model has been tried as well as a cascade classifier as discussed in section 3.1.1, where the ratio of passed tests to remaining tests is interpreted as the confidence. The color based approach experienced similar problems as the mean shift algorithm and is thus not used in this work. The cascade classifier based approach also was inaccurate: The classifier reached relatively high values at regions showing no faces, meaning its output is not representing the confidence of a face being shown at a given position well enough. Therefore, this approach has also not been included in the remainder of this work.

Instead of tracking a face, a method of recognizing faces that have been detected in preceding frames is used. To be able to recognize a face in a later frame, points with distinctive features are selected at the region where the face initially was detected. Those points are easy to track with optical flow and do not suffer from the problems experienced with the face tracking methods described above. In the following is explained how those prominent points can be detected and tracked. In section 4.4 will be described how this approach can be used to identify faces from earlier detections.

3.2.1 Optical flow

The optical flow between two images is the relative displacement of any object observed by the camera or any other observer. In computer vision there is a distinction between *dense optical flow* and *sparse optical flow*. Dense optical flow computes the displacement based on every pixel of the image, for example by using a grid. One popular method for calculating the dense optical flow is described in more in detail in [18]. Sparse optical flow rather computes the optical flow of a selected point. In this work, sparse optical flow is used to track faces through several frames.

Sparse optical flow is used to calculate the displacement vector $\mathbf{d}$ of a given sub window between two subsequent images. A sub windows is used, because a single pixel value might not be unique in its neighborhood. When including a small window around a given location, it is less likely that the pattern is repeated. In order to find the optical flow between two points, different algorithms exists. Here, the *Pyramidal Implementation of the Lucas Kanade Feature Tracker* is used, as described in [19], due to the availability of the robust implementation in *OpenCV*.

The points that are to be tracked must have specific characteristics for the algorithm to work, which is having a gradient within their neighborhood in the vertical as well as the horizontal direction. In [20] a method is proposed that selects such points by finding features that have a distinct gradient, which is implemented in OpenCV as well.

3.3 Face recognition

In this section, an approach to face recognition based on facial feature localization from [21] is explained. This approach first detects facial features and than extracts local appearance based descriptors at where the features were found. When facial features are located successfully, this approach can better deal with different face orientations and transformations.

The facial features are located by combining an appearance model with a part-based "pictorial structure" model. Assuming the appearance of a feature is independent of its positions and the appearance of other features,

"[...] the confidence in an assignment F of positions to each facial feature can be written as a likelihood ratio

$$P(F|\mathbf{p}_1,\ldots,\mathbf{p}_n) = p(\mathbf{p}_1,\ldots,\mathbf{p}_n|F)\prod_{i=1}^{n}\frac{p(\mathbf{a}_i|F)}{p(\mathbf{a}_i|\bar{F})} \tag{3.1}$$

where $\mathbf{p}_i$ denotes the position of feature i in the detected face region and $\mathbf{a}_i$ denotes the image appearance about that point. The joint position of the features $p(\mathbf{p}_1,\ldots,\mathbf{p}_n|F)$ is modeled as a mixture of Gaussian trees. The likelihood-ratio of the appearance terms is modeled using a discriminative classifier" ([21]).

In [21], nine facial features are used, which are the inner and outer corners of the eyes, the left and the right corners as well as the center of the tip of the nose and the left and right corners of the mouth. For each facial feature a binary classifier is learned by using multiple instance AdaBoost learning. The multiple instance variant of AdaBoost can update the location of the labels in the training set in order to prevent localization errors corrupting the classifier ([22]). The output of the classifier is interpreted as the log-likelihood ratio and substitutes $\frac{p(\mathbf{a}_i|F)}{p(\mathbf{a}_i|\bar{F})}$ in equation (3.1). The joint positions of the features are represented by a Gaussian mixture model which is learned by Expectation Maximization and tested with distance transform methods [21].

The descriptor representing the face is build by extracting pixels from an elliptical region around each feature. To reduce the effect of pose variation, the face is transformed so that the features are located at the positions of an average frontal face. The extracted pixels for each feature are first normalized to have zero mean and unit average and then are concatenated. The distance between two descriptors is represented by their Euclidean distance.

3.4 Bag of visual words

Bag of Visual Words (BoW) is a technique from image classification were an image is represented by a histogram of so called visual words. In order to generate a codebook containing the visual words, local features are extracted from all images and are combined to visual words by some clustering algorithm. An overview of the algorithm is shown in figure 3.4, taken from [23]. Which features to use and the choice of the clustering algorithm are up to the user and strongly depend on

the situation. In [1] BoW is used to recognize clothes. Based on this example, the following will explain the details of the BoW approach, as well as the choices made in [1] regarding their specific implementation.

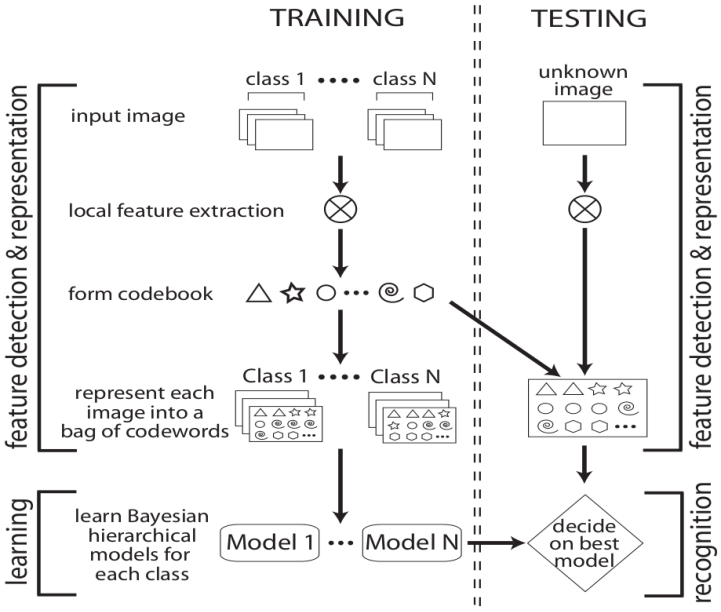


Figure 3.4: Visualization of the Bag-of-Words algorithm, taken from [23]

In [1] clothes of people are detected as described in section 4.5.2. Every detection is represented as a rectangular part of the whole image, containing the clothes of a person. In order to use clothes as an indicator for the identity of a person, a similarity measure between two detections is needed. An simple approach is to just build a histogram of all colors contained by a detection. Although this will work for clothes with distinct colors, it omits any information about the structure and texture on the clothes.

Instead of using the color values, small patches are extracted from all detected clothes. These patches are approximately 7 by 7 pixels and will overlap about 3 pixel. These are the values that have been used by [1], but since no experiments with different values are shown, experimenting with other settings might improve the performance. Each patch is transformed into one vector by concatenating the color values and the pixels after each other. With the settings given here, each patch is thus represented by 147 values. Those vectors represent the local features from the BoW approach.

Before clustering the extracted features, principle component analysis is applied. This has the advantage of removing noise and reducing the length of the features

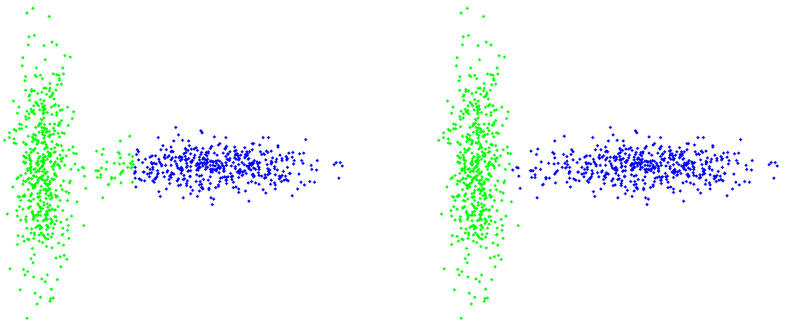
during clustering. For the next step, the generation of the codebook, the 15 first principle components of the original vectors are used.

To all these features represented by their first principal components k-means clustering is applied. The resulting clusters build the codebook for the BoW. In [1] the Mahalanobis distance is used as the distance measure in k-means which is defined as follows:

$$D_M = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{3.2}$$

where x is a vector of values $x = (x_1, x_2, \ldots, x_n)^T$, μ is a vector of means $\mu = (\mu_1, \mu_2, \ldots, \mu_n)^T$ and S is the covariance matrix.

Compared to the Euclidean distance, the Mahalanobis distance returns small distances for points that have a high Euclidean distance but also a high standard deviation. As a consequence, k-means is able to restore clusters with variances. figure 3.5 visualizes the results of k-means with the two different distance measures.



(a) k-means clustering using Euclidean distance

(b) k-means clustering using Mahalanobis distance

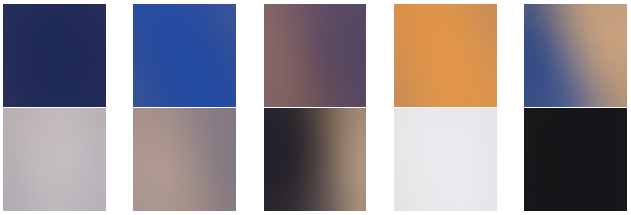
Figure 3.5: K-means applied to a dataset consisting of two different distributions with different deviations.

In figure 3.6 the back-projected visual words are shown for different settings.

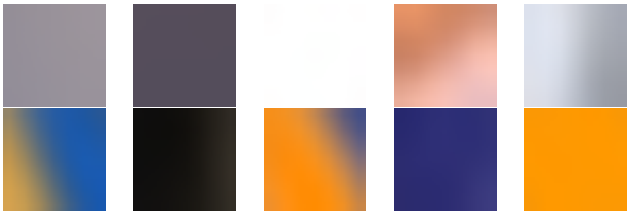
To finally get a descriptor for detected clothes, each patch is assigned to its nearest cluster. The assignment is also done in PCA space, just as the clustering. For each detection a histogram of cluster references is build, or in other words a histogram of visual words.

(a) 5 Visual Words with L2 distance

(b) 5 Visual Words with Mahalanobis distance



(c) 10 Visual Words with L2 distance



(d) 10 Visual Words with Mahalanobis distance

Figure 3.6: Back-projected Visual Words for different settings

3.5 Spectral clustering

Spectral clustering is very different to other clustering algorithms like *Expectation Maximization* or *k-means* because the clustering is done in a subspace of the original input data. Basically, spectral clustering does a principal component analysis of the affinity matrix of the dataset and than performs k-means clustering on the first K principal components, where K is the number of clusters. This has the advantage, that clusters are built based on the distance of the data points between each other, and not based on the location in original input data space. [24] gives an overview of different approaches to and detailed information about spectral clustering.

In spectral clustering, the dataset that is to be clustered is represented as a similarity graph. Given a dataset with data points $X_1, X_2, \ldots, X_n$ and a similarity measure $s(X_i, X_j) \geq 0$, a similarity graph $G = (V, E)$ can be constructed from a dataset as follows:

- Each data point $X_1, \ldots, X_n$ becomes a vertex V of graph G.
- Between every two vertices V_i, V_j an weighted edge is added, with the weight equal to the similarity of the two vertices $w_{i,j} = s(X_i, X_j)$.

The weight matrix W of this graph is defined as the matrix $W = (w_{ij})_{i,j=1,2,\ldots,n}$, containing the pairwise similarities. Furthermore, the degree matrix D of graph G is the diagonal matrix containing the degree of all vertices $v \in V$. The degree of a vertex in a weighted graph is defined as $d_i = \sum_{j=1}^{n} w_{i,j}$.

A graph with weight matrix W and the degree matrix D can be represented as a single matrix, called its Laplacian. There are different definitions of the Laplacian, one of those being the symmetric normalized Laplacian defined as

$$L_{sym} = I - D^{-1/2} W D^{-1/2}, \tag{3.3}$$

which is used in the clustering algorithm in [1].

Given k, the number of clusters to build, and the matrix L_{sym}, the dataset can be clustered as follows:

- Compute the first k eigenvectors $v_1, \ldots, v_k$ of L_{sym}.
- Create a matrix $T = \Re^{n \times k}$ that contains the eigenvectors $v_1, \ldots, v_k$ as columns.
- Row normalize T so that $\sqrt{\sum_{j=1}^{k} t_{i,j}^2} = 1 \quad \forall i \in (1, \ldots, n)$.
- Cluster the rows of T using k-means.

4 Approach

The goal of this project, as explained in section 1.2, is to detect any social interactions that are observable in a video from egocentric perspective. The detected interactions are to be visualized in a time based index, to offer a visual summary of long lasting video material.

In this chapter, first the requirements to the algorithm to be developed are pointed out. Second, an overview of the different stages of the approach is given. Following the overview, the three stages *detection*, *tracking* and *clustering* are explained in detail. Last, in section 4.6, several extensions to the methods on which this work is based are presented.

4.1 Requirements

In order to create a visual index of interactions in egocentric video data (as sketched in figure 1.2), first, the program is required to detect interactions in general. Furthermore, for every detected interaction, the following data is required:

1. Visual representation of the interaction partner
2. The frame index at which the interaction is observed for the first time
3. The frame index at which the interaction is observed for the last time

Given this data, it is possible to create a visual index of the people that were interacted with.

4.2 Algorithm overview

The algorithm is divided into three parts, *detection*, *tracking* and *clustering*. The input and output to and from the three parts is shown in figure 4.1.

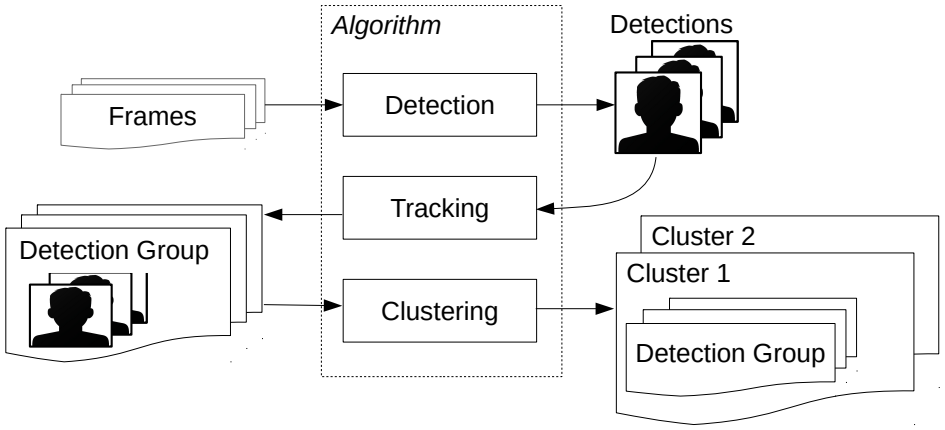


Figure 4.1: Algorithm Overview

The first part, *detection*, is responsible for detecting interactions in each frame of the video. The output are so called *detections*, indicating if and where people are detected in the different frames. This part is explained in more detail in section 4.3.

Second, in *tracking*, the algorithm uses face tracking in order to recognize consecutive detections of the same individual, resulting in *detection groups*. The *tracking* phase is described in section 4.4.

Finally, in *clustering* the *detection groups* are clustered, so that each cluster contains all *detection groups* that represent the same individual. Opposed to *detection groups*, a cluster is expected to contain the complete set of detections belonging to an interaction with someone. That means, the set of *detections* within a cluster is suitable to extract the first and the last frame in which a particular individual occurred. Hence, the required information about when an interaction had begun and when it had ended are contained within a cluster. *Clustering* is dealt with in section 4.5.

4.3 Detection

To detect faces within single frames of a video a cascade classifier as proposed in [17] is used. This technique is explained in more detail in section 3.1. By requiring a certain minimum detection size it is possible to filter people at higher distances as they are more unlikely to be interaction partners.

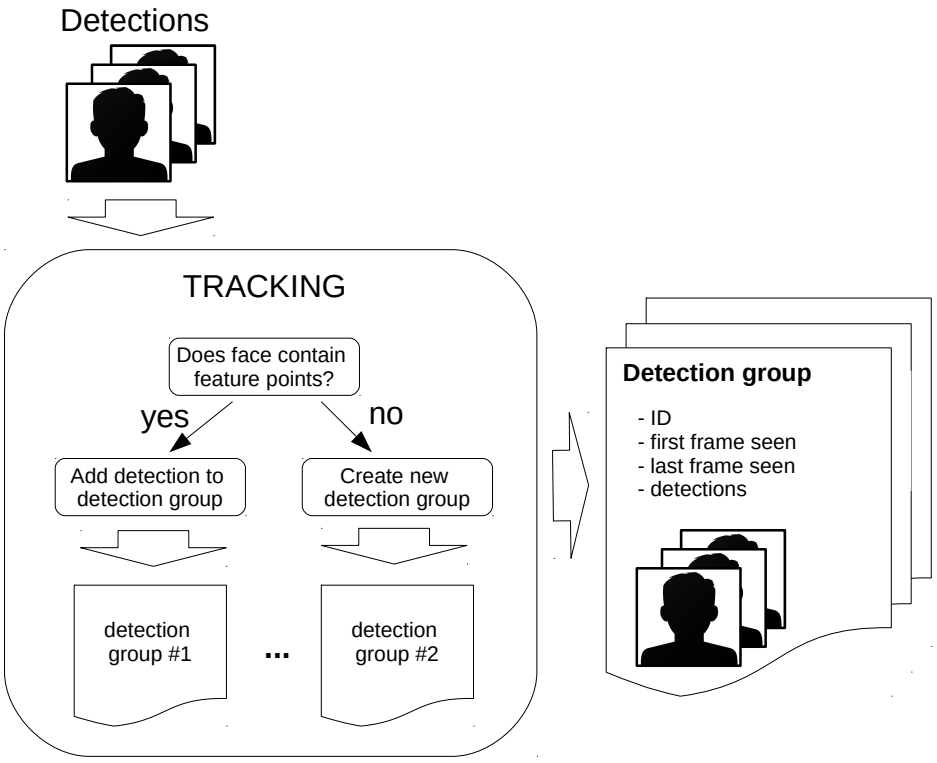


Figure 4.2: Build detection groups from tracking

Although [17] can detect frontal faces as well as profile faces, in this approach only frontal faces are searched for. Profile faces cannot easily be recognized by face recognition techniques and therefor they do not add information required to construct a person based index.

4.4 Tracking

This section explains how faces can be tracked through several frames by using optical flow. An overview of the detection and tracking part is depicted in figure 4.2.

For each new detection it is first checked if it belongs to a person that is already tracked by the tracking module. If a face is not tracked yet, within the rectangular area which is returned from the face detection method, a couple of feature points as described in section 3.2.1 are sampled and stored together with the detection.

In the following frames, these points are tracked using optical flow, also described in section 3.2.1.

The feature points will move together with the face, as they are translated according to the optical flow between the frames. Due to changes in light, as well as rotation or partial coverage of the head, some points can be lost during tracking. Therefore, for each detection, the number of associated feature points can decrease over time. If a faces leaves scene, all points are lost immediately.

Any new detection is considered to be tracked, if a minimum number of feature points from another already tracked detection lie within the new detection. If the detection belongs to a tracked interaction, the new detection will be added to the interaction. Different detections that are known to represent the same individual from tracking, are called *detection groups*. As it turned out during the progress of this work, the same approach for tracking faces has been used in [21].

4.5 Clustering

As discussed, detection and tracking of faces will produce a set of many short interactions, so called *detection groups*, which usually last about several seconds. In order to give a clear summary of such interactions, it must somehow be detected which of these segments belong to the same interaction and which do not. To accomplish this, an extended version of the clustering approach from [1] is used.

The clustering framework proposed by Song was meant to sort a set of photos according to the people that are shown on them. Given a photo album containing 100 photos of five different people, the algorithm can successfully detect the individuals on the photos and is able to cluster most detections into bins containing the same person. The goal of [1] was to help people to order their photo albums and find pictures of the same person quickly.

In order to recognize whether two persons are the same individual or not the algorithm uses both face recognition and clothes recognition. Both techniques provide a distinct measure of similarity between two detections and are combined to single scalar value before spectra clustering is applied.

The problem of clustering photos showing the same person is quite similar to the problem that needs to be solved in order to provide a timeline representing social

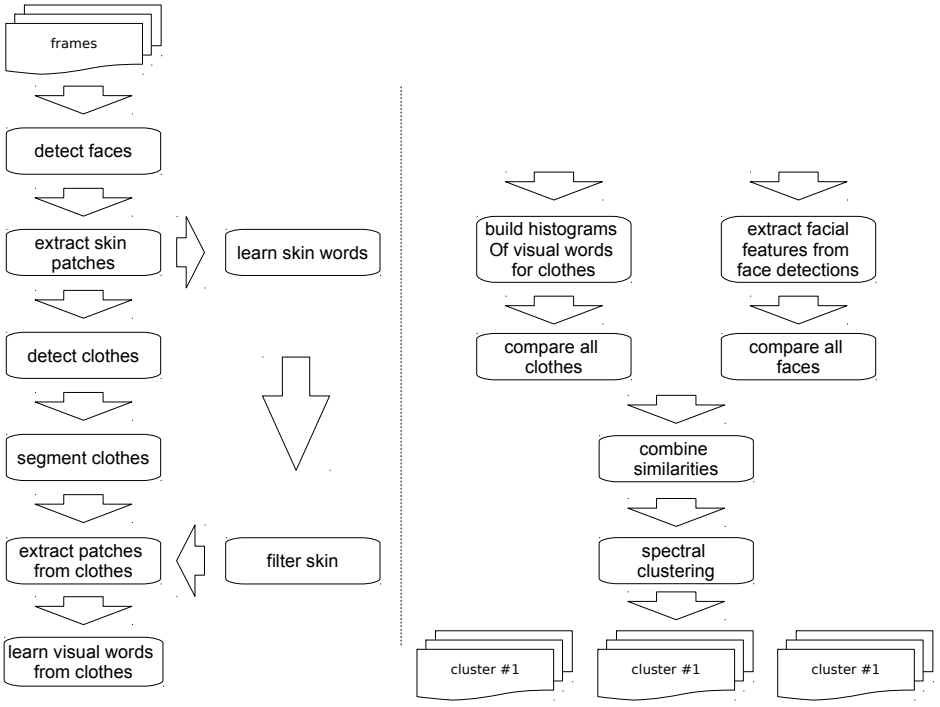


Figure 4.3: Clustering Pipeline

interactions: Some algorithm is required that can decide whether two detections represent the same person or not. That is why in this work [1] is adapted and extended to proposed approach to video indexing.

In the following, first an overview of the clustering algorithm from [1] is given. Afterwards, in section 4.5.2 it is explained how clothes are detected. Next, the feature extraction and comparison with respect to clothes recognition is discussed in section 4.5.3. section 4.5.4 deals with the combination of the distinct similarity measures from clothes and face recognition. Last, in section 4.5.5, the clustering procedure is shown.

4.5.1 Overview

In figure 4.3 an overview of the clustering framework from [1] is shown. The figure shows the original approach, which was designed to process sets of photos. In order to adapt the framework for this approach, in this implementation the face detection step has been replaced with the detection and tracking phase described in section 4.4.

4.5.2 Clothes detection

For every detected face, the clothes descriptors are calculated based on the area below the detected face. While faces usually have not much overlap, or else will not be detected by the face detection module, clothes can often be covered partly by people standing near to each other. By simply drawing a rectangle as just described, such overlap will result in clothes of other people being wrongly included in the descriptor of the detected person. To prevent this issue, [1] applies a simple method to segment the clothes after the initial detection: For all detected clothes on the same image, histograms of the colors are built. The rectangles are shifted and scaled within a predefined range in order to maximize the difference of the histograms.

4.5.3 Clothes recognition and skin detection

All detected clothes are described by a histogram of visual words, as described in section 3.4. The steps *Extract clothes patches*, *Learn visual words for clothes* and *Build histograms of visual words for clothes* from figure 4.3 deal with the construction of such descriptors.

While the general process is based on the typical bag of visual words approach one extension has been made in [1]: While clothes can help to distinguish between different people, often parts of skin will overlap with the parts of an image where clothes are detected. For example, when someone wears a t-shirt, her arms would also be considered clothes. As a result, skin will be included in the codebook of visual words and also in the histograms. Since skin color of different people often is similar, this decreases the ability of the resulting descriptors to distinguish between different clothes. To prevent skin from corrupting the descriptors, patches that are assumed to be skin are neither used for the generation of the codebook nor be included during generation of the histograms.

To be able to decide whether a given patch is part of someones skin or her clothes, a skin detector is used which also is based on the bag of words approach. The codebook for this skin detector is learned by detecting eyes with a cascaded haar filter classifier as described in section 3.1 and taking patches from below the eyes. Patches are collected from all detected persons on all images. In contrast to the word generation for clothes where PCA was used to filter noise, for skin patches simply the mean of all color channels is taken. Each patch will thus result in

three numbers, the mean value for each color channel. The codebook for skin is than generated by applying k-means to all those average color values of the patches, clustering them into ten clusters.

Each patch that is extracted from the clothes needs to fulfill the following two constraints to be considered skin: First, the variance of illumination of the given patch must be below a certain threshold. Second, the minimum distance between the mean of each color channel of the current patch and one of the skin words must be below a threshold, too. The first condition models the fact that skin is usually smooth and illuminated uniformly, in opposite to textiles, which often have wrinkles. The second condition ensures that only those patches which have a similar color than one of the learned skin words are considered to be skin.

After all descriptors are calculated, each bin of each descriptor is reweighted by being multiplied with $\log(\frac{1}{w_i})$, with w_i being the fraction of all patches that has been assigned to the ith bin of all descriptors. This adjustment emphasizes rare patches as they contain more information compared to patches that are frequent in all detections.

4.5.4 Similarity measure

Each detected individual is described by two descriptors: A histogram of visual words representing the clothes and a vector normalized pixel values extracted from the detected feature points as explained in section 3.3.

The similarities between the faces and the clothes of two detections can be measured separately, but for the clustering step a single value describing the similarity between two individuals is required. In [1], linear logistic regression is used to combine the two descriptors.

Linear regression is used to predict a single target variable by linearly combining one ore more input parameters:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \tag{4.1}$$

Linear regression can model continuous target variables well, but is not suitable to express binary variables: The target variable represents the similarity between two individuals and should be between one (same person) and zero (different persons). When using equation (4.1) it is not guaranteed that its outcome will

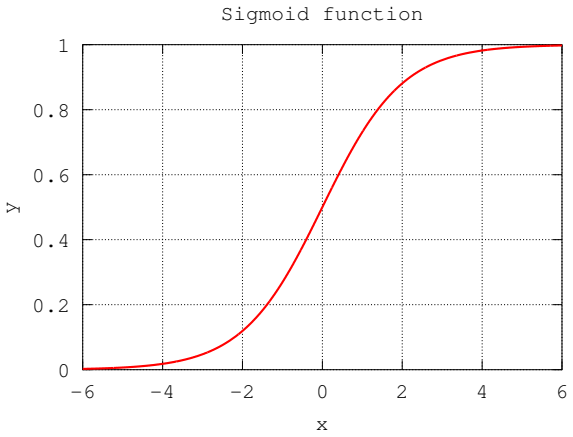


Figure 4.4: Sigmoid function

be between zero and one. Any out-liner in the data can result in higher or lower values.

To force the model to produce results between zero and one the logistic sigmoid function, shown in figure 4.4, is used:

$$y(\mathbf{x}, \mathbf{w}) = \frac{1}{1 + exp(w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n)} \tag{4.2}$$

The probability that two detections are the same person is given by

$$P(Y = 1|x_f, x_c) = \frac{1}{1 + exp(-w_f x_f - w_c x_c - w_0)}, \tag{4.3}$$

where x_f is the similarity between the faces and x_c is the similarity between clothes. The weights w_f and w_c control how much influence each of the similarities has on the combined outcome, and w_0 provides an offset. Given a labeled training set, the values for w_0, w_c and w_f can be learned by applying iterative reweighted least squares. In this work the values has been chosen experimentally due to the lack of an appropriate data set.

4.5.5 Spectral clustering

After having explained the detection and comparison of individuals, the last step required to identify photos or frames of videos that show the same individual is to cluster the resulting detections by similarity. Each resulting cluster represents one individual and all detections contained by one cluster should show the same

person. For this step spectral clustering is used, which is explained in section 3.5. As shown, spectral clustering requires an affinity matrix, containing all pairwise similarities between all detections. To construct the affinity matrix, the similarity measure given in section 4.5.4 is used. Based on this affinity matrix, spectral clustering will return a set of K clusters, where K is required to be given.

4.6 Video specific extensions

From the possibility of tracking people through several frames, prior knowledge about the individual's identity is gained. In [1], no possibility to integrate such prior knowledge is given. In this work, two different approaches are made to integrate the knowledge that is gained from tracking.

One approach is to build detection group specific dictionaries of visual words. This allows the dictionaries to include visual words that are specific to a single individual. When using a global dictionary, the visual words represent the most frequent and most distinct features that are found regarding the complete dataset. Individuals with a similar appearance are likely to be represented by the same visual words as in k-means the extracted patches will have a small distance to each other and thus are likely to be assigned to the same cluster center. Using detection group specific dictionaries instead would lead to both individuals being represented by their own local visual words. Furthermore, this approach allows to calculate dictionaries on-line, not requiring the recording to be finished before starting with the calculation of the descriptors. The resulting descriptors can of course not be compared to each other without including the visual words into the measure of distance. In section 4.6.1, different approaches to including the dictionaries into the distance measure between the descriptors of clothes are shown.

Another approach to integrate the additional information from tracking is to build a statistical model of the appearance per detection group. By including information about the deviation of the appearance within one detection group, the descriptors should be able to better scope with occluding objects and changes in illumination. Furthermore, when representing the detection groups by a single descriptors, the number of descriptors to cluster can be reduced significantly, compared to having a descriptor per detection. The approach to modeling descriptors that include information from multiple detections is described in section 4.6.2.

4.6.1 Multiple dictionary bag of words

When using separate dictionaries per *detection group*, it is not possible to compare descriptors to each other directly when they refer to different dictionaries. In the following, three different approaches are presented that can solve this issue by integrating the dictionaries into the distance measure between two descriptors. All of the following approaches do not support the global reweighting to emphasize rare patches that is described in section 4.5.3, because the bins do not reference a global dictionary anymore.

Separate dictionaries with earth mover's distance

The first possibility is to change the distance measure used to compare the descriptors. Instead of using the dot product between two descriptors, the Earth Mover's Distance (EMD) can be used, which allows to take into account the distance between the dictionaries as well. The EMD can be though of the minimum amount of work that is required to fill holes in the ground with earth from piles in some distance to the holes.

In [25], the EMD between two signatures $P = \{(p_1, w_{p1}), \ldots, (p_n, w_{pn})\}$ and $Q = \{(q_1, w_{q1}), \ldots, (q_n, w_{qn})\}$ is defined as

$$\mathbf{EMD}(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}, \tag{4.4}$$

where $\mathbf{D} = [d_{ij}]$ is the ground distance matrix, with d_{ij} holding the distance between p_i and q_j, and $\mathbf{F} = [f_{ij}]$ being the flow matrix, holding the flows between weights w_{pi} and w_{qj} that minimize the overall cost

$$WORK(P, Q, \mathbf{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}, \tag{4.5}$$

subject to

$$\begin{aligned} f_{ij} &\geq 0 && 1 \leq i \leq m, 1 \leq j \leq n \\ \sum_{j=1}^{n} f_{ij} &\leq w_{pi} && 1 \leq i \leq m \\ \sum_{i=1}^{m} f_{ij} &\leq w_{qj} && 1 \leq j \leq n \\ \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} &= \min(\sum_{i=1}^{m} w_{pi}, \sum_{j=1}^{n} w_{qj}) \end{aligned} \tag{4.6}$$

Intuitively, applying the EMD to compare descriptors of visual words can be explained as follows: When comparing two descriptors, the visual words of the dictionaries they refer to represent the location of the piles and holes that are to be filled. The visual words of the first descriptor represent the location of the piles and the visual words of the second descriptor represent the location of the holes that are to be filled. The distance between two visual words is the difference in their appearance. As the descriptors being histograms of visual words, their entries represent the size of the holes and the weight of the piles. Consequently, high values in the histograms can only be *moved* to the other descriptor at low cost when the visual words they represent have a similar appearance and similar size.

Formally speaking, the first part of the signature, $p_1, \ldots, p_n$, are the back-projected visual words. The visual words are back-projected because each *detection group* uses an individual PCA transformation vector. As a result, the principal components cannot be compared with each other in PCA space, because the principal components of each group have a completely different meaning in original space. The second part of the signature, the weights $w_1, \ldots, w_n$, are the histograms of visual words. The ground distance matrix $\mathbf{F}$ is calculated by taking the L1-distance between every two visual word vectors p_i and q_j. By normalizing the histograms to sum up to 1, all descriptors have the same amount of weight and the EMD between them becomes a metric.

Multiple dictionaries bag of words unified dictionaries

Another possibility to deal with local dictionaries of visual words is to combine the dictionaries of different descriptors and recalculate the descriptors based on

the combined dictionary. The resulting descriptors can be compared with each other since they refer to the same combined dictionary. In [26], two different approaches of merging dictionaries are introduced. One possibility is to merge different dictionaries into one dictionary and recalculate the histograms based on the new dictionary, called multiple dictionaries bag of visual words with unified dictionaries (MDBoWUD).

For example, given two *detection groups* A and B, when comparing descriptors from *detection group* A to descriptors from *detection group* B the steps are as follows: A new dictionary AB is built including all visual words from dictionary of *detection group A*, as well as all visual words of *detection group* B. All descriptors that are to be compared from *detection groups* A and B are rebuilt based on the combined dictionary AB. The descriptors are built, as described in section 3.4, by assigning each extracted patch to the closest visual word from dictionary AB. The resulting histograms refer thus to the same dictionary and can be compared without regarding the underlying visual words. This approach, together with MDBoW with separate dictionaries, is visualized in figure 4.5.

Compared to the approach from section 4.6.1 where the EMD is used to measure the distance between the visual words, this has the advantage of not requiring to compare the visual words to each other. Hence, this method is expected to be more exact and better be able to distinguish different individuals. A disadvantage is an decrease of performance, as for every possible combination of detection groups, all descriptors have to be recalculated.

Multiple dictionaries bag of words separate dictionaries

Another approach to multiple dictionaries from [26] is multiple dictionaries bag of visual words with separate dictionaries (MDBoWSD). Instead of merging two dictionaries, for each detection a second descriptor is built, based on the other descriptor's dictionary. The detection is then described by a new concatenated descriptor, containing the two individual histograms. The resulting descriptors reference both dictionaries and thus are comparable. This approach is also visualized in figure 4.5.

The main difference to the approach of merging dictionaries is that now each patch is represented by two bins, referring two the different dictionaries. By having to calculate one half of the descriptor only, the performance of this method is better compared to merged dictionaries.

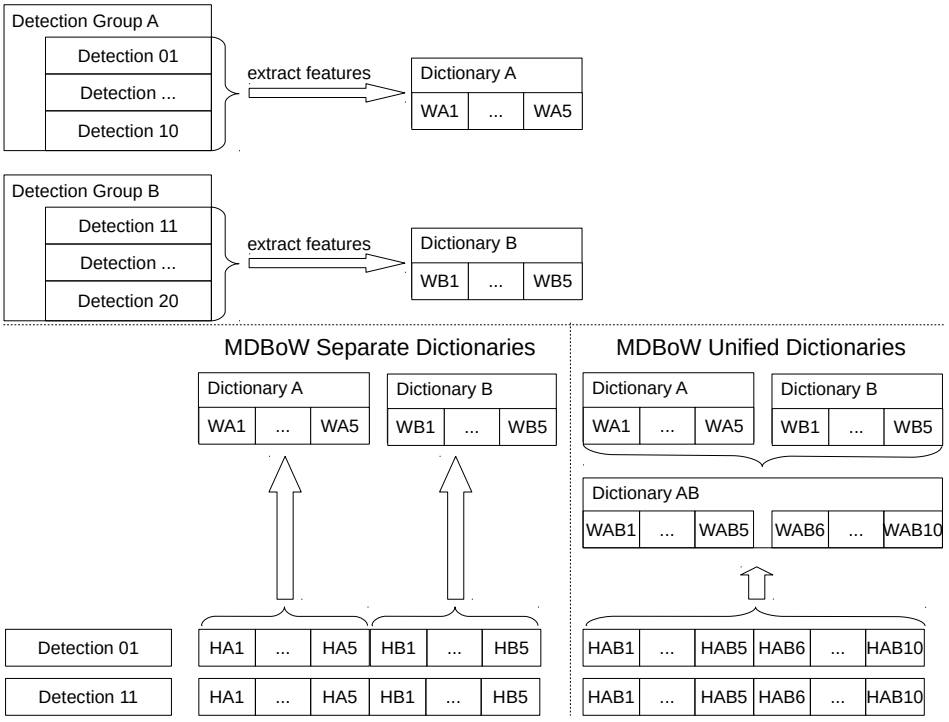


Figure 4.5: Multiple dictionaries bag of words approaches

4.6.2 Descriptors from multiple detections

In the original approach by [1], the appearance of each detection is modeled as a histogram of visual word frequencies. To build a descriptor that contains information of more than one of such descriptors, this work evaluates a way of modeling multiple histograms as a single descriptor.

The most simple and straight forward approach is to build a vector containing the average values of each bin. The advantage of this approach is that the effect of outlining values is reduced significantly. On the other hand, bins with a high variance within one detection group will not be represented appropriately, as the information about the variance is lost.

A better approach is to use two vectors containing not only the average, but also including the standard deviation for each histogram bin. In opposite to histograms or a vector of average bin values, the values of each bin cannot be compared by simply subtracting the values. Instead, each bin is interpreted as a Gaussian distribution and the distance between two bins is the probability of them describing each other. For this, the normalized L2 distance between to Gaussian distributions is used, which is described in [27] as

$$d_{nL2}(p_1, p_2) = \int (p_1'(x) - p_2'(x))^2 \mathrm{d}x, \tag{4.7}$$

where

$$p_i' = p_i(x) / \sqrt{\int p_i(x)^2 \mathrm{d}x} \tag{4.8}$$

In the implementation, $\int p_i(x)^2 \mathrm{d}x$ is approximated by sampling 1000 linear data points between 0 and 1.

5 Evaluation

In this chapter, the different approaches proposed in chapter 4 are evaluated experimentally. In the following, the data that is used for the evaluation is described in section 5.2. Next, in section 5.3, the setup of the experiments is explained. In section 5.4, the results of the experiments are given, and the discussion can be found in section 5.5.

5.1 Datasets

For the experimental evaluation, two different datasets are used. The first dataset consists of a set of photos and is used to compare the results of the implementation to the results from [1]. The second dataset consists of egocentric video data and is used to evaluate the performance of the proposed framework with respect to the project goals. In the following, both datasets are described in more detail.

5.1.1 Dataset 1

Dataset 1 is selected to be comparable to the data that has been used in the evaluation of [1]. With this dataset, the results of the implementation of the parts *detection* and *clustering* are compared to the results of [1].

In [1], a set of photos of 3-10 people has been used, annotated with the date on which the picture was taken. When the pictures were taken on the same day, clothes recognition was used in [1]. To imitate the characteristics of the data, dataset 1 is assembled from different press photos taken during the coronation of the dutch king Willem Alexander on April 30th, 2013: The photos show three different individuals (Willem Alexander, his wife and his mother) with a prominent visual distinction. To prevent other individuals than the three mentioned from being included in clustering, all other faces were blurred and hence not detected by face detection. Clothes recognition can be used for all photos from dataset 1 since they were taken on the same date and the individual's appearance

does not change. The dataset contains 15 photos showing three different individuals, as listed in table 5.1. From such a small set with clearly distinguishable appearances, the algorithm is expected to cluster the detections with very high accuracy.

Individual	#1	#2	#3
Detections	12	10	4

Table 5.1: Characteristics of dataset 1.

5.1.2 Dataset 2

For the purpose of evaluating the clustering performance of the proposed framework with respect to video material, a new video dataset has been collected. First, a set of 6 interactions with cashiers of different shops were recorded. The interactions were than concatenated sequentially to create a test video. The video material was recorded with the camera of a mobile phone, mounted in a chest pocket, as in the beginning of the project no head mountable camera was available. The recordings show characteristics very similar to videos recorded with a head mounted camera.

The resulting clips are summarized in figure 5.1. From the video, detections and detection groups were extracted as is described in section 4.3. The number of detections and detection groups resulting from applying face detection and tracking the detected faces with optical flow are shown in table 5.2.

Individual	#1	#2	#3	#4	#5	#6
Detections	59	201	52	44	81	66
Detection Groups	3	3	3	2	3	5

Table 5.2: Characteristics of dataset 2.

5.2 Evaluation measures

To evaluate the proposed framework, two methods are used to measure its performance. First, receiver operating characteristics (ROC) is used to measure the quality of the clusters resulting from applying the proposed approached. section 5.2.1 describes how the ROC are extracted from the results as well as how it can be interpreted with respect to the project's goal. Second, k-nearest neigh-

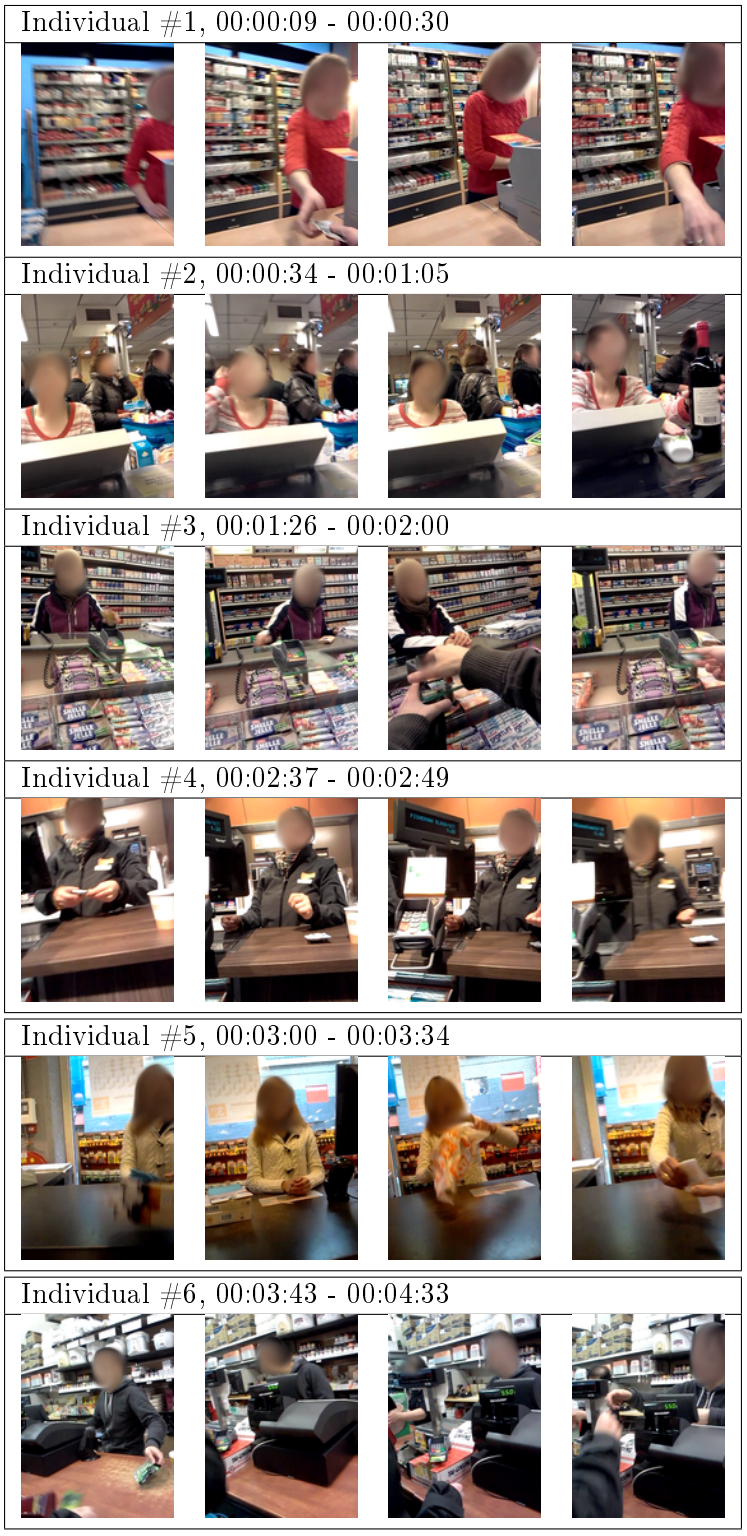


Figure 5.1: Looxcie 2 dataset part 2

bor analysis (k-NN) is used to compare the ability of the different descriptors to distinct between the individuals of dataset 2. section 5.2.2 explains how k-NN is applied in more detail.

5.2.1 Receiver operating characteristics

The receiver operating characteristics curve (ROC curve) visualizes the trade-off between the true positive rate (TPR) and the false positive rate (FPR). In this project, ROC curves are used in order to evaluate the quality and usability of the clustering part of the proposed framework.

With respect to the index that this projects aims to create, this trade-off would reflect as follows: The higher the TPR is, the more overview the index can offer. That is because the higher the TPR, the more detections are assigned to the same correct interaction. This is important because the first frame and the last frame found in one cluster are used to determine when an interaction has started and when it ends. Vice versa, given a low TPR, many detections that belong to the same interaction will be classified as different interactions, leading to a cluttered index.

On the other hand, the FPR represents the fraction of detections that is incorrectly assigned to a interaction they do not belong to. A high FPR can results in a wrong indication of when an interaction starts or ends, or interactions not being displayed by the index at all, for example when all detections of at least two individuals are contained by a single cluster.

In clustering, the TPR and the FPR can be measured by counting the number of correct and incorrect decisions made during cluster assignment. To calculate the fractions, the following terms are used:

- True Positives (TP) is the number of correct decisions to put two items into the same cluster.
- False Positives (FP) is the number of incorrect decisions to put two items into the same cluster.
- True Negative (TN) is the number of correct decisions to put two items into different clusters.
- False Negative (FN) is the number of incorrect decisions to put two items into different clusters.

The TPR and the FPR are defined as follows:

$$\begin{aligned} \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{FPR} &= \frac{\text{FP}}{\text{FP} + \text{TN}} \end{aligned} \tag{5.1}$$

As proposed in [1], the Rand index ([28]) is used to calculate the TPR and FPR: Given a set of N detections,

> "[...] any clustering result can be seen as a collection of $N(N-1)/2$ pairwise decision. A false alarm happens when a pair actually from different individuals, but the algorithm claims they are the same individual. A true positive (detection) is when a pair actually from the same individual and the algorithm also claims so" ([1]).

To evaluate the clustering algorithm, the resulting TPR and FPR values are given for an increasing number of clusters. Although usually the ROC is visualized in a graph, in most experiments the values for TPR and FPR are given tabularly, since many values are closed to each other and are difficult to distinguish in a visual representation.

An optimal algorithm will lead to TPR $= 1$ and FPR $= 0$, when given the correct number of classes. Increasing the number of clusters above the number of classes from ground truth would decrease the TPR, while the FPR would remain 0. The deviation of the experimental results from these values shows how well the implementation performs compared to an optimal solution.

5.2.2 K-nearest neighbor analysis

To analyze the quality of the different bag of words based visual descriptors, the average precision of the first K neighbors is looked at. An appropriate descriptor should have a small distance compared to descriptors of the same class and a high distance to descriptors of other classes. Since all detections from the same detection group are already known to belong to the same cluster, only those detections are considered which are in different detection groups than the original detection, when retrieving the nearest neighbors. Given a set of detections D and the K nearest neighbors of each detection K_D, the average precision p is the

number of correct neighbors divided by the total numbers of neighbors, given by

$$p = \frac{\sum_{d \in D} \sum_{k \in K_D} knn(d,k)}{\sum_D \sum_{K_D} 1} \quad \text{, where} \quad knn(d,k) = \begin{cases} 1 & \text{if same person} \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

5.3 Experiments

Before describing the setup of each experiment beginning from section 5.3.2, section 5.3.1 introduces several parameters which refer to the different approaches to be evaluated, as well as their default values.

5.3.1 Parameter description

The following enumeration lists all parameters that have been changed by either one or more of the experiments:

- Included Descriptor
  This parameter controls which information should be used during clustering: The possible values are *faces*, *clothes* or *combined*. *Faces* means only face recognition as described in section 3.3 is used, *clothes* means only visual appearance as described in section 4.5.3 is used, and *combined* uses a combination of both, as explained in section 4.5.4.
- Visual Word Clustering
  This parameter controls which clustering algorithm is used to build the visual word dictionary used for clothes recognition. The parameter can either be *k-means*, meaning regular k-means clustering is used, or *mahalanobis*, meaning k-means clustering is used but using the Mahalanobis distance to find the nearest cluster center, as explained in section 3.4.
- Number of Visual Words
  This parameter controls the size of the visual words dictionary, used for clothes recognition.
- Visual Word Dictionary
  This parameter controls which kind of visual word dictionary is used for

clothes recognition. The possible values are listed in table 5.3.

Value	Description
global	Build a single global dictionary of visual words. All clothes descriptors refer to the same dictionary. This approach is described in section 3.4.
MDBOW EMD	Build a local dictionary for every detection group. Use the EMD to compare to descriptors referencing different dictionaries. This approach is described in section 4.6.1.
MDBOW SD	Build a local dictionary for every detection group. The clothes descriptors are built by concatenating the histograms of each dictionary. This approach is described in section 4.6.1.

Table 5.3: Possible values for parameter *Visual Word Dictionary*.

The default value of each parameter is shown in table 5.4.

Parameter	Original Value
Included descriptor	combined
Visual Word Clustering	mahalanobis
Number of Visual Words	30
Visual Word Dictionary	global

Table 5.4: Parameter values used in the original approach

5.3.2 Experiment 1

This experiment is designed to compare the results of the implementation that has been written for this project to the results of the approach from [1]. The *clustering* part is evaluated by clustering dataset 1, and measuring the resulting ROC. The experiment shall show whether the implementation works as expected, given a simplistic dataset. The performance of each possible combination of descriptors is measured separately, which are either descriptors for clothes, descriptors for faces, or a combination of both. This shall show the performance of each descriptor individually, as well as the performance of the combined descriptor.

5.3.3 Experiment 2

The second experiment measures the clustering performance of the original approach when given detections from dataset 2. The results of this experiment are

the benchmark for the evaluation of the extensions proposed in this work. The performance of descriptors for clothes only, faces only, as well as descriptors for a combined distance measure are tested individually.

5.3.4 Experiment 3

As proposed in section 4.6.1, an approach to multiple dictionaries of visual words based on the EMD is implemented, MDBOW EMD. In this experiment, the clustering performance of MDBOW EMD is measured. As discussed in section 4.6.1, reweighting the histogram bins is not applicable when local visual word dictionaries are used, and is thus turned off.

5.3.5 Experiment 4

Another approach to local dictionaries is MDBOW SD, as explained in section 4.6.1. As in experiment 3, in order to determine the performance of the new clothes descriptors as well as the influence on the combined distance, the clustering results when using clothes descriptors and the combined descriptors both are measured. The clustering results are compared to those of the original approach.

5.3.6 Experiment 5

While the experiments 2, 3 and 4 measure the resulting clustering performance of different descriptors for clothes, this experiment is designed to evaluate the ability of the descriptors to retrieve other detections of the same class. Therefor, for each detection in *Dataset 2*, the first 10 most similar detections that do not belong to the same detection group are retrieved, the so called nearest neighbors, as described in section 5.2.2. Opposed to the prior experiments, these results are independent of the clustering algorithm used. A high precision rate in retrieving nearest neighbors while measuring low detection rates in clustering could indicate that the currently used clustering algorithm is not suitable for the given task.

5.3.7 Experiment 6

In experiment 6, the performance of regular k-means clustering is compared to the performance of the original approach, which is using spectral clustering. Opposed to building a global distance matrix as in spectral clustering, in k-means the descriptors are interpreted as multidimensional data points. The combination of the different descriptors of faces and clothes is done by just concatenating both vectors.

5.3.8 Experiment 7

Another new approach, as described in section 4.6.2, is to build descriptors per *detection group*, calculated from the data of all detections within that group. In this experiment, the resulting clustering performance of these descriptors is measured. Again, the individual outcomes for using solely clothes descriptors or face descriptors, as well as the combined descriptors are given. The resulting performance is compared to the original approach.

5.3.9 Experiment 8

While the analysis of the preceding experiments allows for a quantitative measure, the given numbers do not let conclude if an approach will be able to create an suitable index of a given video. In this experiment an actual timeline is produced to demonstrate the degree of usability of the presented approach for summarizing video material.

As one can see from the following section, the quantitative results from the experiment 7, shown in section 5.4.7, suggest that the settings used in that experiment have the lowest false alarm rate while maintaining a high precision. Consequently, in this experiment descriptors per *detection group* will be used.

The number of clusters is set to 8 in this experiment. This decreases the chance that an individual is not shown in the timeline due to an false assignment of more than one distinct individual to a single cluster. Since only one representative detection is shown per cluster, in such case one individual would remain unseen. In general one can say that the higher the number of clusters, the more entries the timeline will show, making the timeline less clear. On the other hand, an increased number of clusters reduces the chance of false assignments of different

individuals to the same cluster.

5.4 Results

5.4.1 Experiment 1

In figure 5.2, the results of clustering dataset 1, described in section 5.1.1, are presented as an ROC curve. For the correct number of clusters, $C = 3$, the precision is 1 and no false assignments appear.

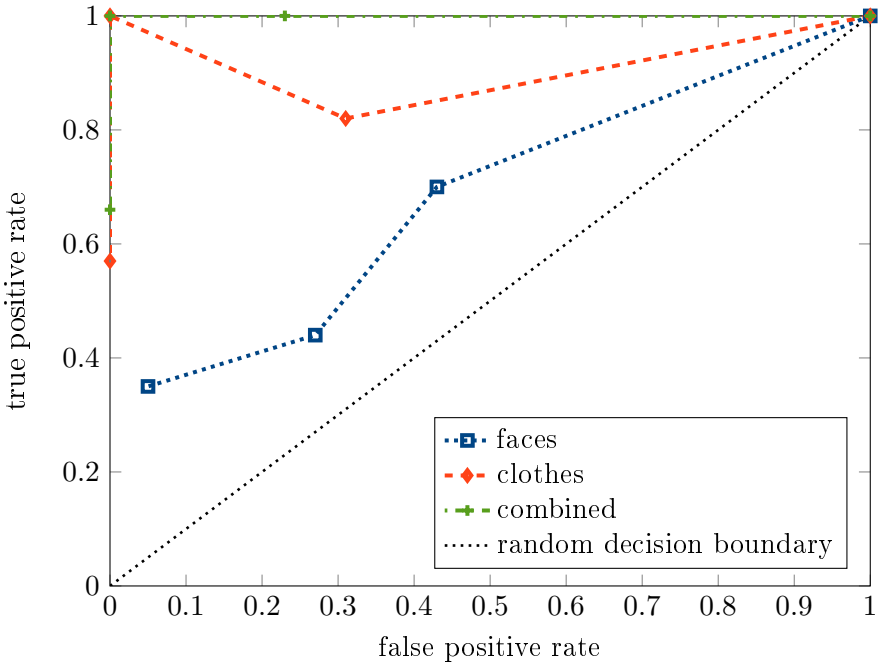


Figure 5.2: ROC curve comparing clothes, faces and combined descriptors. The ROC curve for the number of clusters being 1, 2, 3 and 4, whereas 3 is the correct number of clusters. The point in the top right represents the results for a single cluster. From right to left the number of clusters increases.

5.4.2 Experiment 2

The results of clustering dataset 2 with the original approach are shown in table 5.5.

For each descriptor type, the resulting cluster assignment is shown. The resulting assignment for face descriptors only is shown in figure 5.3, for clothes descriptors

Approach	C=2		C=4		**C=6**		C=12		C=18	
	P	**F**	**P**	**F**	**P**	**F**	**P**	**F**	**P**	**F**
Orig Cloth.	0.99	0.72	0.99	0.43	0.98	0.43	0.97	0.42	0.93	0.11
Orig Faces	0.98	0.49	0.94	0.08	0.63	0.05	0.39	0.00	0.27	0.00
Orig Comb.	1.00	0.73	0.99	0.28	0.98	0.28	0.98	0.29	0.97	0.29

Table 5.5: Experiment 2: The resulting precision (P) and false alarm rate (F) of the original approach. The correct number of clusters is 6.

only in figure 5.4 as well as for combined distances in figure 5.5.

From the cluster assignment for face descriptors only, as shown in figure 5.3, can be seen that the individuals 3, 4, 5 and 6 are correctly assigned to separate clusters, with nearly all detections of those individuals being in the correct cluster. Wrong assignments are found for individuals 1 and 2, as both individuals are split into at least two clusters, whereas at least one third of the detections wrongly being assigned to a separate clusters.

In comparison, when using clothes descriptors only as shown in figure 5.4 or integrating both distances as shown in figure 5.5, the algorithm wrongly assigns most detections of individuals 2, 3, 4 and 6 to a single cluster. Nonetheless, for individuals 1 and 5, most detections are assigned to the correct clusters 2 and 3. When using the combined distance measure, also individual 4 is assigned correctly to a separate cluster. As a result, the precision for clothes descriptors only as well as for a combined descriptors is better, since the detections of individuals 1 and 2 are not divided into several clusters. On the other hand, more detections of different individuals are assigned to a single cluster, leading to an increase of false alarms compared to using face descriptors only.

Although it is difficult to see from the figures, several single detections of individuals 4 and 5 are assigned to incorrect clusters as well. In figure 5.6, detections of individual 4 from different clusters are shown, in order to visualize the appearance of those detections that are assigned correctly and those detections that are assigned incorrectly. For individual 5 samples are shown in figure 5.7.

5.4.3 Experiment 3

The results of clustering dataset 2 using the *MDBOW EMD* approach are compared to the results of the original approach in table 5.6. For the correct number of clusters $C = 6$, the MDBOW EMD approach has a 21% lower FPR than the

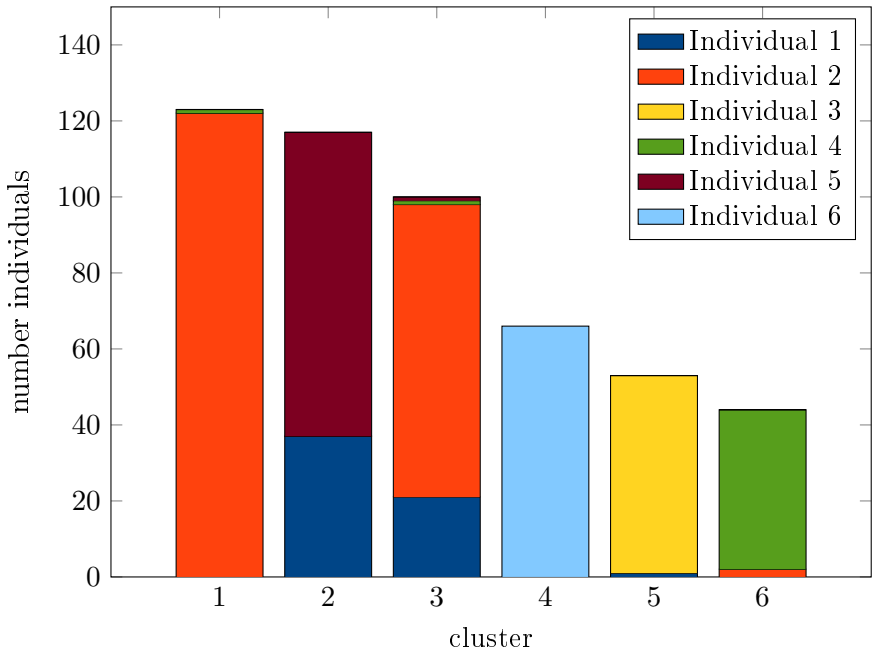


Figure 5.3: Experiment 2: The resulting assignment of individuals to clusters when using face descriptors only, applied to dataset 2.

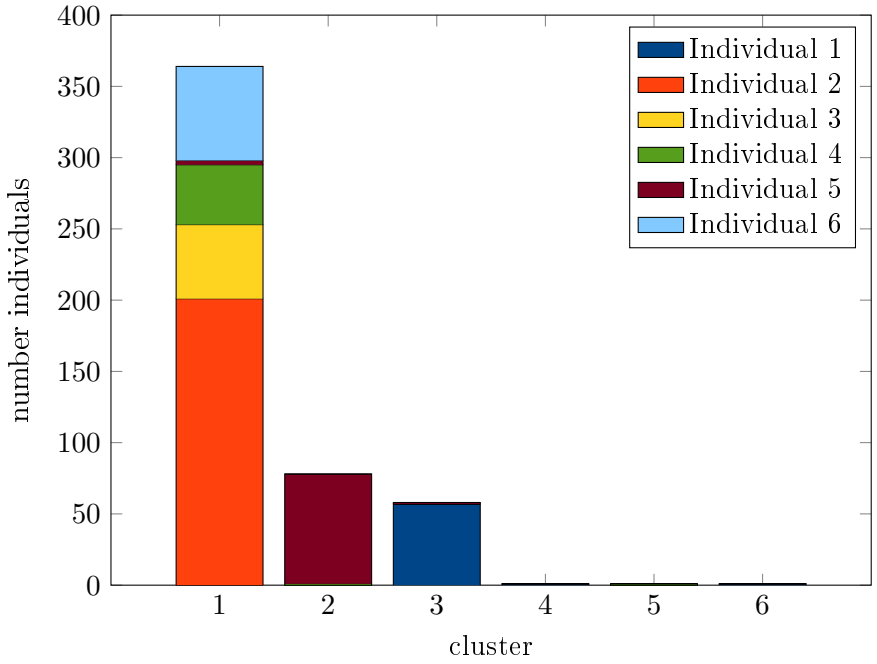


Figure 5.4: Experiment 2: The resulting assignment of individuals to clusters when using clothes descriptors only, applied to dataset 2.

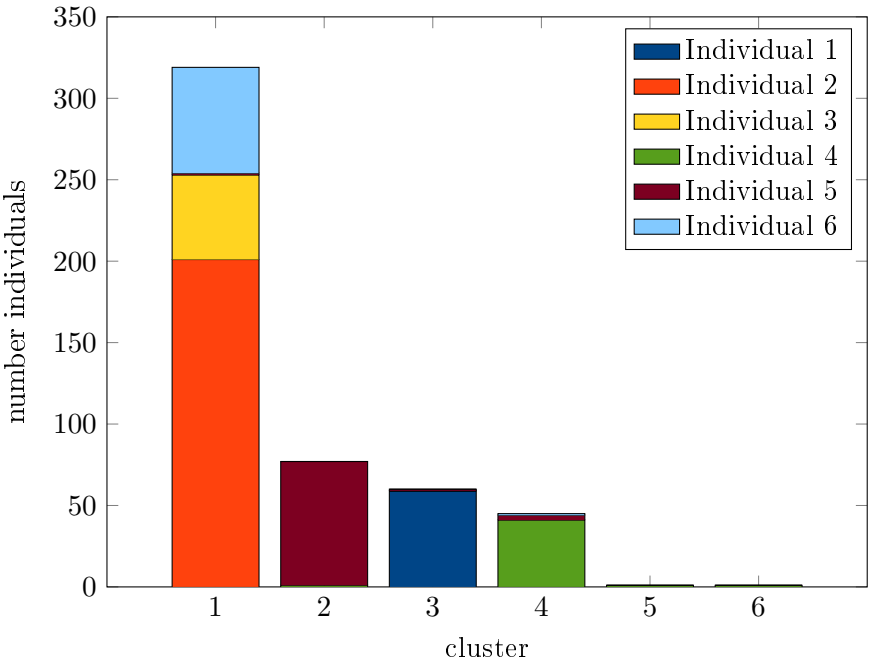


Figure 5.5: Experiment 2: The resulting assignment of individuals to clusters when using a combined distance measure, applied to dataset 2.

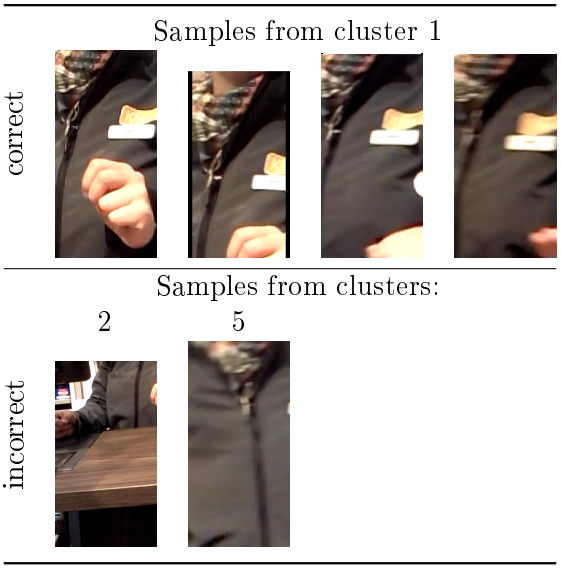


Figure 5.6: Experiment 2: Outliers of individual 4 from clustering based on clothes.

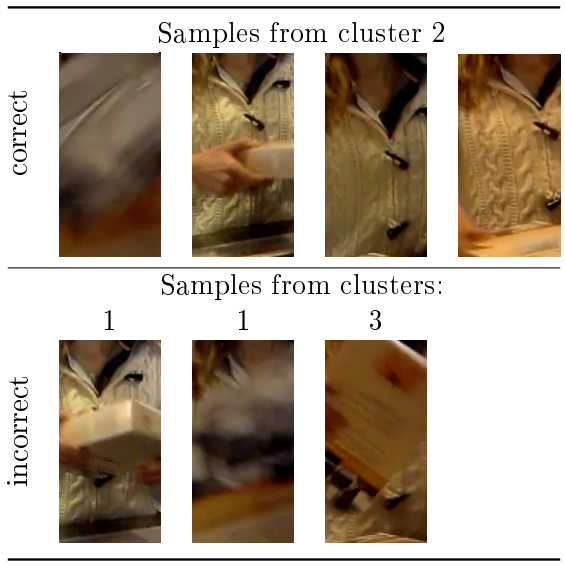


Figure 5.7: Experiment 2: Outliers of individual 5 from clustering based on clothes.

Approach	C=2		C=4		**C=6**		C=12		C=18	
	P	**F**	**P**	**F**	**P**	**F**	**P**	**F**	**P**	**F**
EMD Cloth.	0.99	0.48	0.95	0.23	0.94	0.22	0.98	0.47	0.92	0.22
EMD Comb.	0.99	0.47	0.94	0.22	0.93	0.22	0.98	0.47	0.96	0.46
Orig Cloth.	0.99	0.72	0.99	0.43	0.98	0.43	0.97	0.42	0.93	0.11
Orig Faces	0.98	0.49	0.94	0.08	0.63	0.05	0.39	0.00	0.27	0.00
Orig Comb.	1.00	0.73	0.99	0.28	0.98	0.28	0.98	0.29	0.97	0.29

Table 5.6: Experiment 3: The resulting precision (P) and false alarm rate (F) of the MDBOW EMD approach, compared to the original approach. The correct number of clusters is $C = 6$.

original approach, while the TPR only decreases about 4%. When the detections are clustered into $C = 12$ clusters, the FPR of the same approach is 62% higher, which show the importance of having the correct number of clusters.

The resulting cluster assignment is shown in figure 5.8 and figure 5.9. In opposite to the cluster assignment of the original approach, the *MDBOW EMD* approach fails to create clusters with only one individual. Instead, the majority of the detections of individuals 1 and 5 is assigned to clusters 2, while individuals 3 and 6 are assigned to cluster 3. Although clusters 2 and 3 contain more detections from different distinct individuals when compared to the results from experiment 2, the decreased maximum number of distinct individuals in a single cluster lets the false alarm rate decrease.

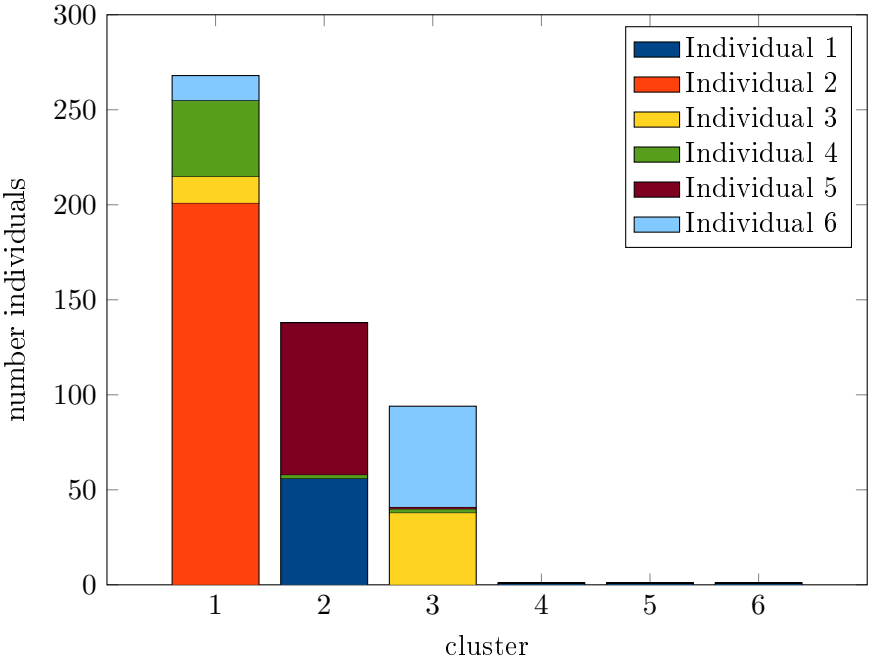


Figure 5.8: Experiment 3: The resulting assignment of individuals to clusters when using clothes descriptors only, applied to dataset 2.

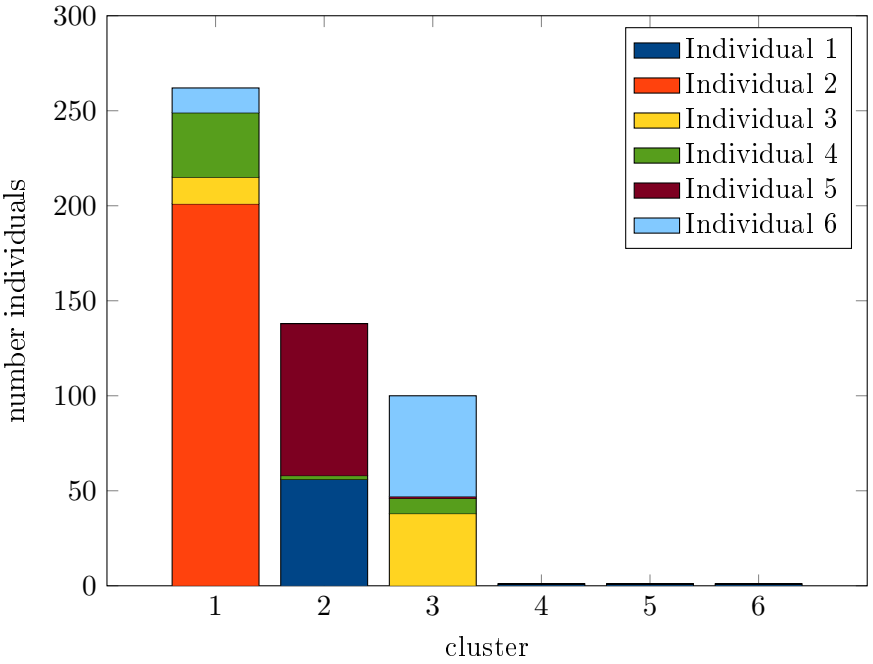


Figure 5.9: Experiment 3: The resulting assignment of individuals to clusters when using a combined distance measure, applied to dataset 2.

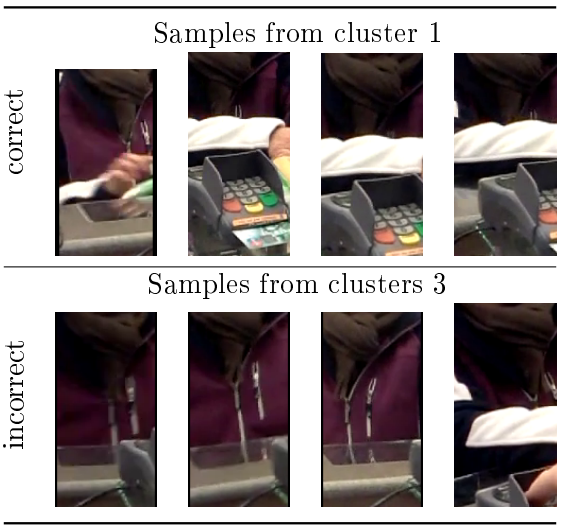


Figure 5.10: Experiment 3: Samples from individual 3 from clustering based on clothes.

To investigate possible causes for the incorrect assignment of individual 3 to the two different clusters 1 and 3, samples of the individual from both clusters are shown in figure 5.10. Samples from individual 4 and 6, also taken from clusters 1 and 3, are shown in figure 5.11 and figure 5.12.

5.4.4 Experiment 4

The results of clustering dataset 2 using the *MDBOW SD* approach are compared to the results of the original approach in table 5.7. When using clothes descriptors only, the precision as well as the false alarm rate of the original approach and the *MDBOW SD* achieve approximately the same values when the number of clusters matches the number of individuals. On the contrary, the resulting false alarm rate of the combined measure for the *MDBOW SD* approach is noticeably higher than the value of the original approach, while the precision is still about the same in both cases.

In figure 5.13, the assignment of the samples to the different clusters is shown for the *MDBOW SD* approach using clothes descriptors only. The *MDBOW SD* approach effectively assigns the samples of the 6 individuals to only two clusters in the clothes descriptors experiment: While the original approach is able to distinct individuals 1 and 5, the *MDBOW SD* method assigns both individuals to the same cluster. The outliers of individual 4 that are wrongly assigned to cluster 2, as well as the outliers of individual 5 that are wrongly assigned to

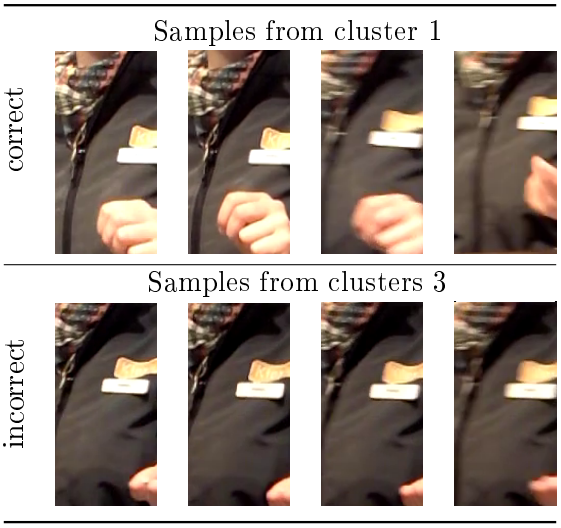


Figure 5.11: Experiment 3: Samples from individual 4 from clustering based on clothes.

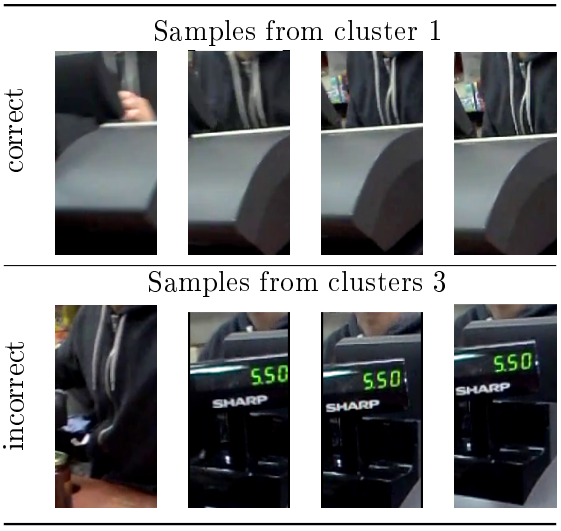


Figure 5.12: Experiment 3: Samples from individual 6 from clustering based on clothes.

Approach	C=2		C=4		**C=6**		C=12		C=18	
	P	**F**	**P**	**F**	**P**	**F**	**P**	**F**	**P**	**F**
SD Clothes	0.97	0.47	0.99	0.47	0.99	0.47	0.97	0.47	0.95	0.46
SD Comb.	0.97	0.46	0.97	0.45	0.96	0.45	0.96	0.44	0.95	0.44
Orig Cloth.	0.99	0.72	0.99	0.43	0.98	0.43	0.97	0.42	0.93	0.11
Orig Faces	0.98	0.49	0.94	0.08	0.63	0.05	0.39	0.00	0.27	0.00
Orig Comb.	1.00	0.73	0.99	0.28	0.98	0.28	0.98	0.29	0.97	0.29

Table 5.7: Experiment 4: The resulting precision (P) and false alarm rate (F) of the MDBOW SD approach, compared to the original approach. The correct number of clusters is $C = 6$.

cluster 1 are the same samples as in experiment 2 when using the combined distance measure. For the visualization of those samples please refer to figure 5.6 for individual 4 and figure 5.7 for individual 5.

The resulting assignment from the *MDBOW SD* approach with a combined distance measure is visualized in figure 5.14. In contrast to the assignment resulting from the clothes descriptors only approach, the samples of the fourth individual are wrongly assigned to multiple clusters, namely cluster 1 and 2. This phenomenon also occurred when applying the *MDBOW EMD* approach. For a visualization of the samples of individual 4 from clusters 1 and 2 please refer to figure 5.11.

5.4.5 Experiment 5

The average precision of retrieving the first 10 nearest neighbors for each detection is shown in figure 5.15. The precision of the original approach for $K = 1$ is 0.82 and decreases approximately linearly to 0.68 for $K = 10$. The *MDBOW EMD* approach and the *MDBOW SD* approach have an overall better precision, which is about the same for both approaches. For $K = 1$ the precision for the two new approaches is about 0.95 and decreases to 0.8.

5.4.6 Experiment 6

In table 5.8, the resulting precision and false alarm rate are shown for the approach of using k-means instead of spectral clustering, compared to the values of the original approach. When using k-means, the resulting false alarm rate is significantly lower. At the correct number of clusters $C = 6$, the false alarm rate

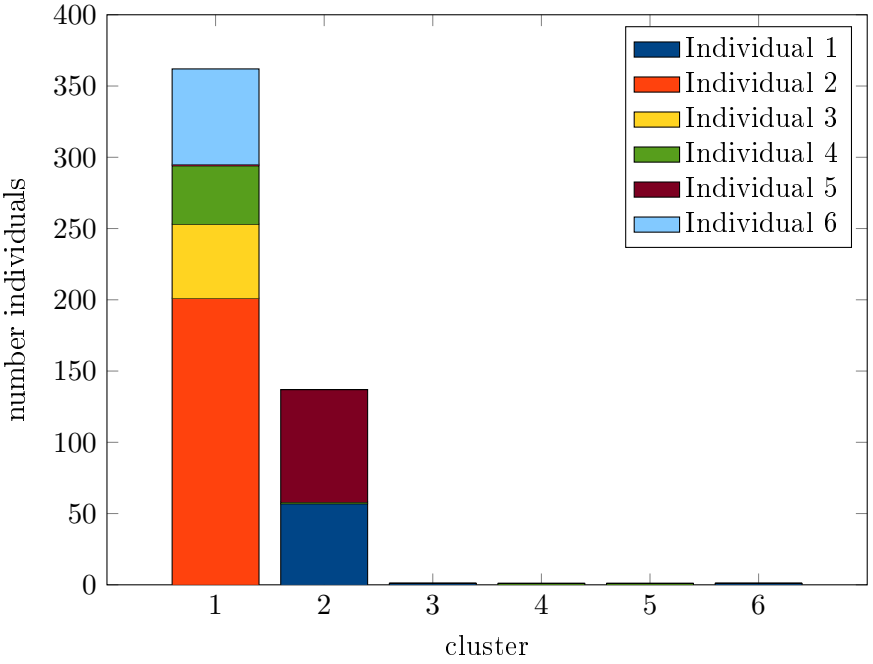


Figure 5.13: Experiment 4: The resulting assignment of individuals to clusters when using clothes descriptors only, applied to dataset 2.

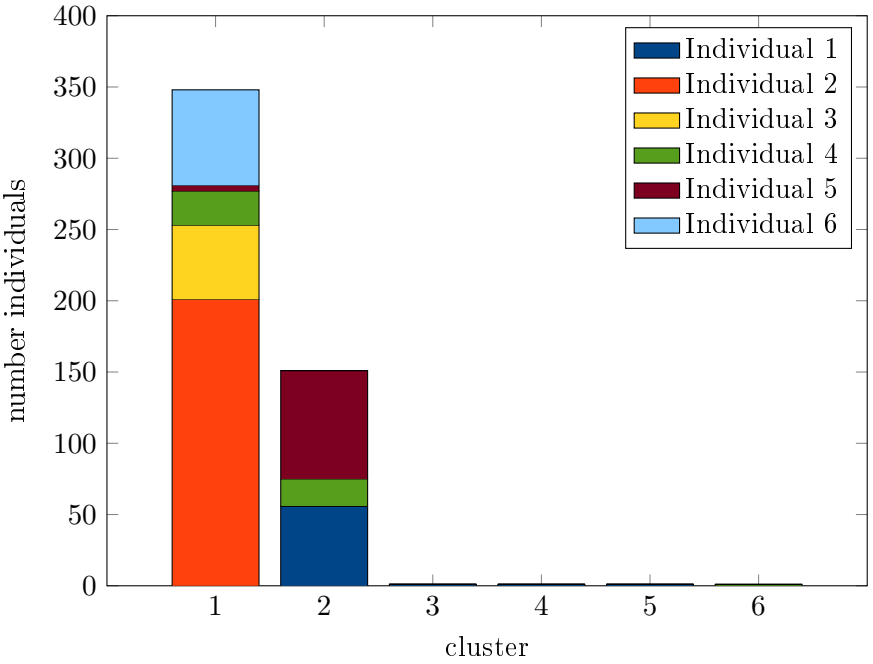


Figure 5.14: Experiment 4: The resulting assignment of individuals to clusters when using a combined distance measure, applied to dataset 2.

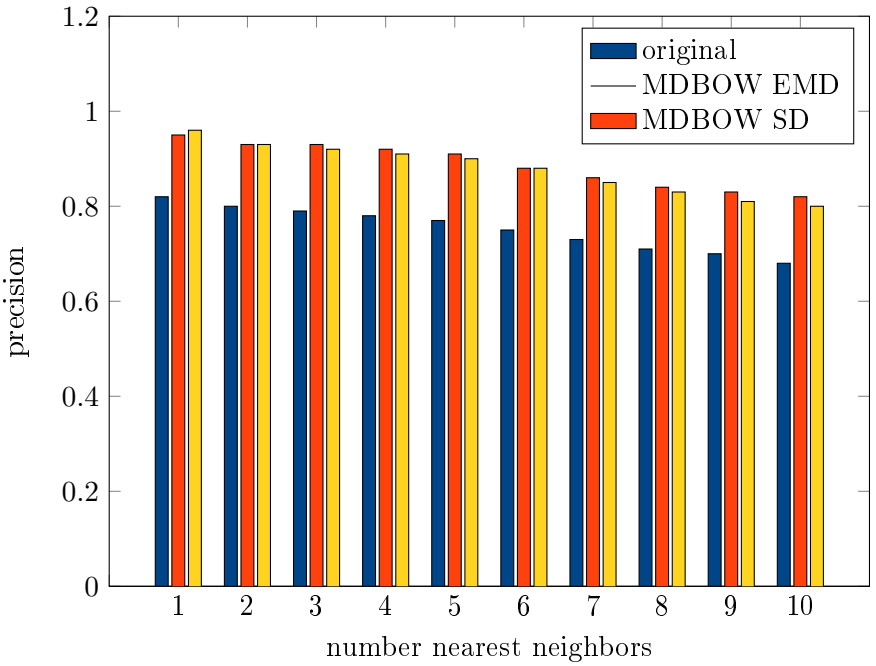


Figure 5.15: Experiment 5: The average precision of different clothes descriptors when retrieving the K nearest neighbors for each detection in dataset 2.

Approach	C=2		C=4		**C=6**		C=12		C=18	
	P	**F**	**P**	**F**	**P**	**F**	**P**	**F**	**P**	**F**
k-means	1.00	0.40	0.92	0.11	0.90	0.04	0.38	0.02	0.30	0.01
Orig.	0.99	0.72	0.99	0.43	0.98	0.43	0.96	0.11	0.97	0.43

Table 5.8: Experiment 6: The resulting precision (P) and false alarm rate (F) of the approach of clustering the descriptors with k-means, compared to the original approach. The correct number of clusters is $C = 6$.

for k-means is 0.04, compared to 0.43 when using spectral clustering. The precision on the other hand is better when using spectral clustering. When building 12 clusters, k-means results in a precision of no more than 0.38, while spectral clustering has a precision of 0.96.

5.4.7 Experiment 7

The results of clustering dataset 2 by using the approach of detection group based descriptors are compared to the results of the original approach in table 5.9. For the correct number of clusters $C = 6$, clustering detection groups results in a TPR decreased by 1% and a FPR decreased by 89%. When increasing the number of

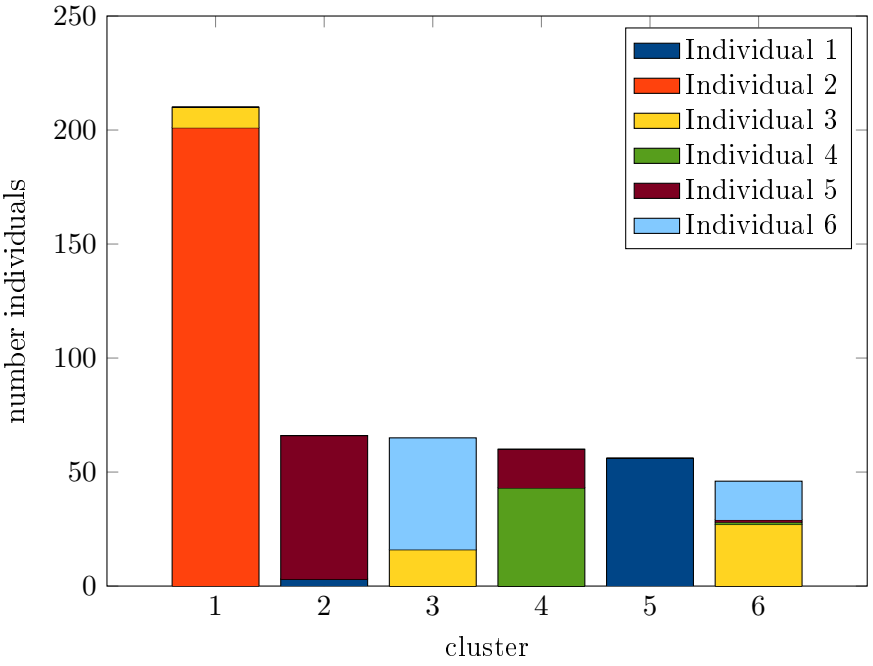


Figure 5.16: Experiment 6: The resulting assignment of individuals to clusters when using clothes descriptors only, applied to dataset 2.

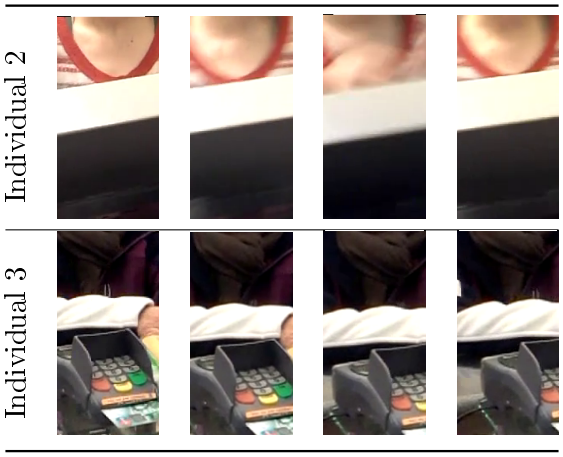


Figure 5.17: Experiment 6: Samples from cluster 1 from clustering based on clothes.

Figure 5.18: Experiment 6: Samples from cluster 3 from clustering based on clothes.

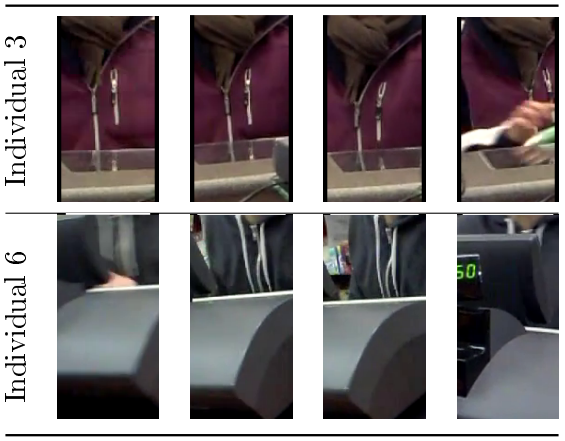


Figure 5.19: Experiment 6: Samples from cluster 3 from clustering based on clothes.

Approach	C=2		C=4		**C=6**		C=8		C=10	
	P	**F**	**P**	**F**	**P**	**F**	**P**	**F**	**P**	**F**
DG Clothes	1.00	0.37	0.95	0.11	0.95	0.12	0.93	0.09	0.93	0.04
DG Faces	0.83	0.91	0.81	0.88	0.54	0.52	0.52	0.41	0.51	0.19
DG Comb.	1.00	0.37	1.00	0.09	0.97	0.03	0.96	0.02	0.95	0.00
Orig Cloth.	0.99	0.72	0.99	0.43	0.98	0.43	0.96	0.11	0.97	0.43
Orig Faces	0.98	0.49	0.94	0.08	0.63	0.05	0.53	0.01	0.42	0.01
Orig Comb.	1.00	0.73	0.99	0.28	0.98	0.28	0.98	0.29	0.97	0.28

Table 5.9: Experiment 7: The resulting precision (P) and false alarm rate (F) of the approach of detection group based descriptors, compared to the original approach. The correct number of clusters is $C = 6$.

cluster to 10, which is 1.6 times the correct number of clusters, the FPR becomes zero and the TPR is 0.95, meaning that there are no clusters containing more than a single distinct individual.

As can be seen from the assignment diagram in figure 5.20, the reason of the decreased false alarm rate is that there are no more outliers in the clusters, with outliers meaning small numbers of detections being assigned to the wrong clusters. Furthermore, the detections of the individual with the highest number of detections, individual 2, are assigned to a separate cluster, reducing the total number of detections in the least homogeneous cluster, being cluster 2.

In contrary, the false alarm rate is considerably higher when using the detection groups based approach with face descriptors only. From the assignment diagram in figure 5.20 can be seen that the algorithm produces wrongly assigns samples from all six individuals to a single cluster, in this case cluster 1.

Nonetheless, when integrating the faces descriptors and clothes descriptors to a combined descriptor, the resulting cluster assignment has a lower false alarm rate than for clothes descriptors only. As can be observed in figure 5.22, individuals 1, 2 and 5 are perfectly assigned to distinct clusters. For individual 6, also most detections are within a distinct cluster, except about 15% of its samples being assigned to cluster 2, together with all detections from individuals 3.

5.4.8 Experiment 8

The ground truth timeline for dataset 2 is depicted in figure 5.23. Each horizontal rectangle represents the presence of the depicted individual within the video.

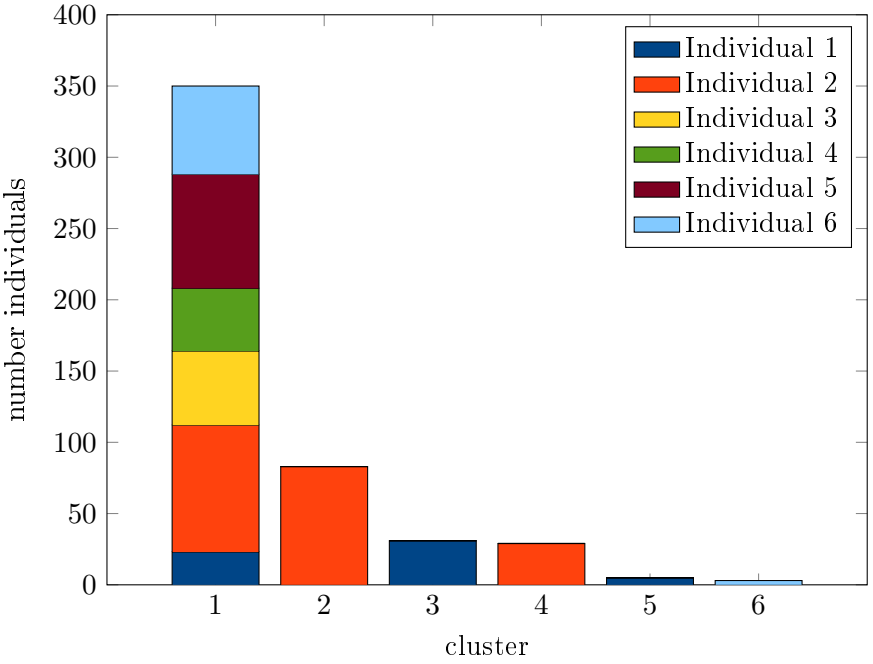


Figure 5.20: Experiment 7: The resulting assignment of individuals to clusters when using face descriptors only, applied to dataset 2.

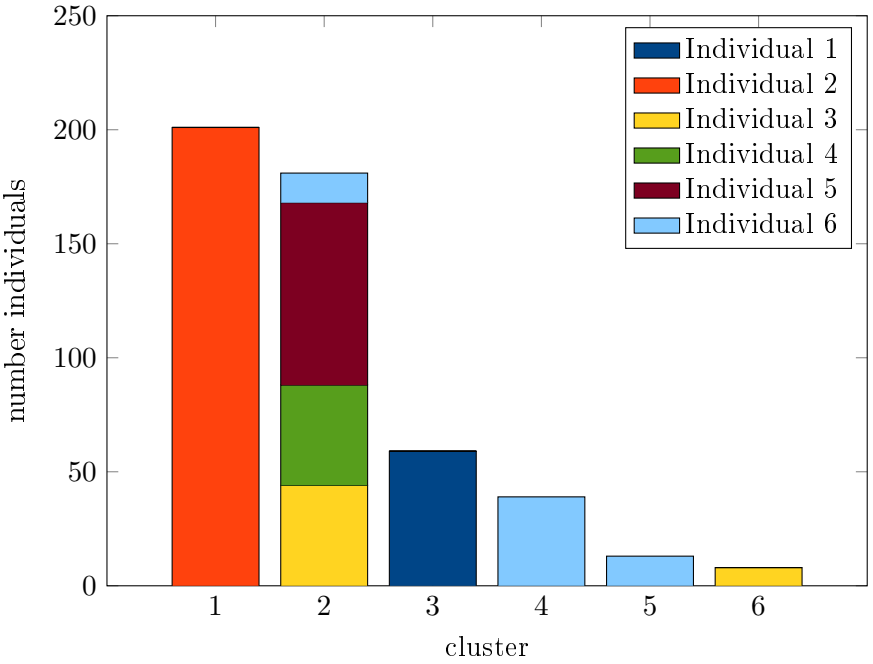


Figure 5.21: Experiment 7: The resulting assignment of individuals to clusters when using clothes descriptors only, applied to dataset 2.

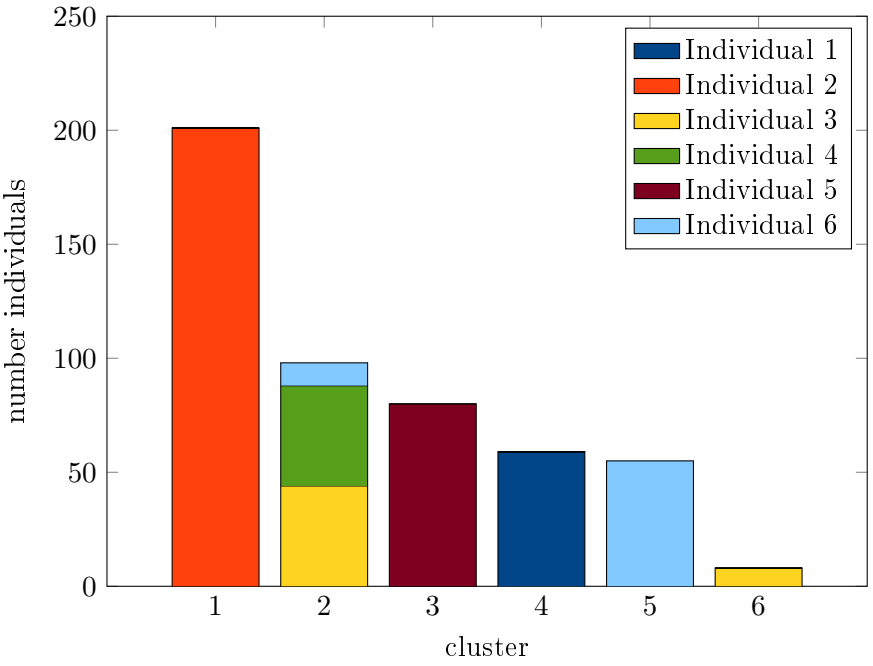


Figure 5.22: Experiment 7: The resulting assignment of individuals to clusters when using a combined distance measure, applied to dataset 2.

In figure 5.24, the timeline is shown that is constructed by the framework. Individuals 1,2 and 5 are detected at the correct position and with approximately the correct number of frames. Individual 6 is detected at the correct position, although clearly several occurrences are not recognized, letting the timeline entry begin later and end earlier than the entry in the timeline representing ground truth. For individuals 3 and 4, two separate timeline entries are shown. While individual 4 is represented correctly, the 3rd individual's last frame is wrongly recognized at the end of the video.

5.5 Discussion

5.5.1 Experiment 1

The results of the first experiment show that the implementation is working according to the expectations: The combined approach results in a perfect clustering of dataset 1. The better results of using clothes descriptors only, compared to using face descriptors only can be explained by the characteristics of dataset 1: The individuals were clothes with a distinctive appearance. That a combination

Figure 5.23: Experiment 8: The timeline produced for ground truth clustering results of dataset 2.

of both descriptors results in more homogeneous clusters confirms the results of [1].

5.5.2 Experiment 2

For the comparison of the generated results with former work, the results of the experiments with the dataset *family 1* from [1] has been used, because the number of seven individuals matches the number of individuals in experiment two the best. In comparison to the results in [1], the detection rate as well as the false alarm rate achieved in the second experiment are about the same when using face descriptors only. When using the combined distance measure, experiment two results in a better detection rate (0.98 in experiment two and 0.65 in [1]) but also more false positive assignments (0.28 in experiment two and 0.05 in [1]).

A possible cause for the good detection rate lies in the nature of the dataset: The detections are taken from sequential frames of a video meaning that the visual changes between consecutive detections are small. Since spectral clustering is based on the connectivity between the data points, consecutive frames are likely to be assigned to the same cluster.

The high false alarm rate on the other hand is presumably caused by objects that temporarily occlude the individuals' appearance. For example, often the objects to be bought are handed to the recorded individual, occluding a major part of the individual's appearance. The same problem occurs during the payment of

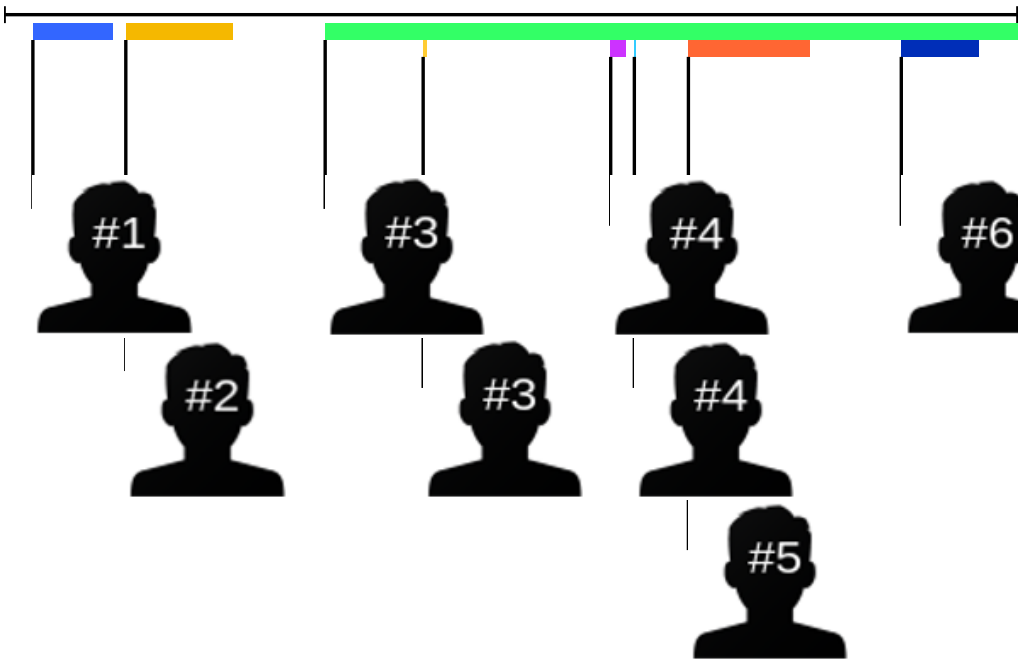


Figure 5.24: Experiment 8: The timeline produced for ground truth clustering results of dataset 2.

the groceries, where frequently the payment terminal overlaps the area detected as clothes. Consequently, these objects are wrongly interpreted as an individual's appearance. As occlusions emerge over multiple frames, the change of appearance per frame is small and such changes are likely to be included in the same cluster as other neighboring frames. In such cases the cluster is not only likely to include detections with a high affinity to the individual's appearance, but also detections with a small distance to the occluding object's appearance. In other words, occlusions can increase the connectivity between detections of different clusters, leading to an increased false alarm rate.

5.5.3 Experiment 3

Detections from one individual that have correctly been assigned to a distinct cluster when using the original approach being assigned to different clusters when using the MDBOW EMD approach can have two causes. First, the visual words are not global but calculated per detection group. Detection groups contain significantly less frames and are likely to include samples of a single individual. If an individual is occluded by an object for a significant number of frames, the visual words are likely to reflect the object's visual appearance. As a consequence, the visual words of two detection groups of the same individual can differ decisively. Hence, the descriptors of two detections from neighboring frames can possibly refer to quite different visual word dictionaries. If one dictionary fails to represent the common visual features with the other dictionary, the EMD will result in a high distance between two otherwise similar detections. Since spectral clustering depends on the linkage between single frames, such differences can cause the algorithm to wrongly assign the detections to different clusters.

The second possible cause of individuals being assigned to wrong clusters is the change of how the distance between two histograms is calculated. Normally, when comparing two histograms h_1 and h_2, the values for each bin of the histograms are compared separately. When using the EMD, values of one histogram can be compared to all bins from the other histogram, taking account the distance between the two bins. This can be of great effect on visually similar detections of two distinct individuals: When the visual features of each individual are captured as distinct visual words during dictionary generation, similar features may correctly be assigned to distinct bins. For example, h_1 will hold the magnitude of the features in bin 0, while h_2 will hold the magnitude of its features in bin 1. Considering usual histogram comparison, the distance between h_1 and h_2 will be high because the values of bin 0 as well as the values of bin 1 show great difference. Contrary the EMD will see the values as weight and the distance between the bins as the work required to move the values between the bins. Thus, when the distance between the bins is short, resulting in a short distance between the two histograms.

Considering these disadvantages, it is questionable if the proposed methods still has any advantage over using simple color histograms.

5.5.4 Experiment 4

The MDBOW SD approach results in less false alarms than the MDBOW EMD approach. The possible reason is that although having local dictionaries per detection group, each histogram bin is still only compared to its corresponding bin within the other descriptor. Thus, the loss of information occurring when using MDBOW EMD does not occur when using MDBOW SD.

Nonetheless, another problem emerges, which can explain why the MDBOW SD approach still results in a higher false alarm rate than the original approach: The descriptors consist of two separate parts which are concatenated, each part based on one of the two dictionaries according to the detection groups. Thus, each feature is assigned to the nearest word twice, once per dictionary. Consequently, when both detections have similar features, as well as similar visual words, both parts of each descriptor are expected to be alike as well, as are the two resulting descriptors. The MDBOW UD approach is thus expected to better distinct between two similar detections, as it assigns each feature to only one of the two dictionaries.

5.5.5 Experiment 5

The average precision in retrieving the first K nearest neighbors is better when either the MDBOW EMD approach or the MDBOW SD approach is used, compared to the original approach. A possible explanation is that the MDBOW approach results in dictionaries that are very specific to a detection group. They can model the appearance of each detection group more precisely since the visual words are generated from detections of the same individual only. Concluding, in case of existing information about subgroups within a dataset, both proposed approaches are attractive choices for a color based descriptor. Nonetheless, they do bring disadvantages has been is discussed previously in section 5.5.3 and section 5.5.4.

5.5.6 Experiment 6

Using k-means instead of spectral clustering results in a decreased false alarm rate but also a decreased precision. While with spectral clustering often clusters are left empty, k-means is better able to divide the detections into 6 distinct

clusters. A possible explanation is that k-means does not regard the connectivity between all frames as does spectral clustering. Instead, detections are assigned to the closest cluster center. As a result, sequences which contains changes in appearance are not recognized by k-means while spectral clustering does.

On the other hand, for the same reason k-means is less frail to the influence of objects overlapping the appearance of the individual. In case of spectral clustering, because of the relatively small change in appearance per frame, occluding objects have strong linkage to their neighboring frames. Next to the neighboring frames, the occluding object has a strong linkage to other detections with a similar appearance. With k-means, the linkage between multiple frames is not considered. Thus the occlusion of an individual's appearance by any object only affects the frames in which that object is visible.

To conclude one can say that spectral clustering can better scope with a change of appearance but is prone to corruption from occluding objects.

5.5.7 Experiment 7

The approach of aggregating detections to *detection groups* results in a significantly lower false alarm rate compared to the other approaches. With this approach the decrease of false alarms does not lead to a decrease of precision either.

In search for reasons of the improvement experienced in this experiment it comes to mind that the number of data is significantly reduced. Instead of 501 detections the algorithm has to assign only 18 detection groups to the correct clusters. Nonetheless, if the chances of making the correct choice are unchanged, the algorithm would still be expected to perform at a similar error rate. Thus, the observed decrease in false alarms must arise from an improved availability of making the correct choice, based on the aggregation of information from multiple detections within a detection group.

Interestingly, the decrease of the false alarm rate within the combined approach is not reflected when using face descriptors only. Still, the combined approach works better than regarding clothes descriptors alone. This demonstrates the benefit of a combination of different kind of descriptors, as can be seen when looking at the assignment diagram in section 5.4.7: In figure 5.20, the assignment diagram for face descriptors only, one can see that individual 1 and individual

2 are wrongly assigned to three different clusters, each. On the contrary, when looking at the assignment diagram for approach using clothes descriptors only in figure 5.21, both individuals are correctly assigned to a single cluster each but instead individual 6 is incorrectly assigned to three clusters. When combining the two descriptors, shown in figure 5.22, those three individuals nearly perfectly clustered into three different clusters, except for several detections of individual 3. Furthermore, even individual 5 is assigned a separate cluster now. As discussed in section 5.5.6, spectral clustering is prone to occluding objects which may lead to inter-cluster linkages, resulting in a high false alarm rate. By combining different descriptors this effect is expected to be of less impact, since the occlusion might affect only one of the descriptors. Also, ambiguities in one of the descriptors are not necessarily found in the other descriptor. For example when two individuals were similar clothes they can still have different facial characteristics.

To conclude, the proposed method of combining individual descriptors into a probability distribution is an attractive choice.

5.5.8 Experiment 8

The generated timeline shows all individuals by picture and all but one individual are detected at the correct position and length. The index thus provides reliable information about who is in the video and can for example be used when searching for a video showing an specific individual.

Furthermore, the provided information can help to give a clue about the content of the video by showing at which time each individual appears. Although the latter information is not correct for one individual, the timeline still can help people to get an approximate impression of who appears when in the video.

6 Conclusion

6.1 Summary

The main goal of this work is to create a framework which can summarize video data based on the individuals appearing in it. The approach taken is to use spectral clustering while integrating face recognition and context information, like the appearance of the individuals, as proposed in [1] regarding photographic material. Next to the method from [1], different extensions that include context information specific to video material are evaluated. Last, it is demonstrated to what extent the proposed framework is able to create a person based index of a short video including the sequential appearance of six different individuals.

6.2 Contributions

The contributions of this work are the proposed extensions to [1] that include video specific context information:

First, with *MDBOW EMD* an approach to multiple dictionaries of visual words by using the earth movers distance is proposed and evaluated. *MDBOW EMD* allows to build dictionaries of visual words from subsets of the complete dataset and allows descriptors referring to different dictionaries to be compared without further recomputation. The experiments show that using multiple dictionaries with the Earth Mover's Distance results in a 21% lower false positive rate than the original approach, while the true positive rate only decreases 4%.

Second, the *MDBOW SD* approach from [26] is evaluated in the context of automated video indexing. In this context the approach to multiple dictionaries by concatenating the descriptors is less effective than the original approach, but the k-NN experiment suggests that both methods perform better than the original approach when used to query for similar detections. In other words, the methods might render very useful in other applications.

Third, a method that integrates prior knowledge by modeling multiple descriptors as Gaussian distributions of bins is presented. When used in clustering, the resulting false alarm rate is considerably decreased, leading to much more homogeneous clusters compared to the original BOW approach from [1]. The false positive rate is reduced by 89%, given the correct number of clusters. With this method the proposed framework is able to automatically render a human interaction based visual index of the dataset that has been recorded in the scope of this work.

6.3 Limitations

The proposed framework for video indexing still has several limitations. As one can see from section 5.4.4, when using descriptors based on detection groups, the algorithm is not able to produce perfect clusters that represent exact one single individual. As a result, the timeline produced by the algorithm does not yet match the ground truth of the dataset, as has been demonstrated in section 5.4.8.

Furthermore, the dataset used for the evaluation of the framework is very limited regarding its structure and size. During the research, no suitable dataset was found available that fulfilled the requirements of being recorded from an egocentric perspective and at the same time showing frequent interactions with other individuals. As a consequence, the dataset for the evaluation has been recorded within the scope of this work with the available time constraining its complexity. The resulting experimental evaluation is thus limited with regards to real world data as well as comparability with other work.

6.4 Future work

Referring to the discussion of experiment 4 in section 5.5.4, implementing *MD-BoW UD* could help to further decrease the false alarm rate within the clusters. Consequently, the timeline generated based on such clustering results is expected to be more precise.

Another direction of research could be to extend the context of the information that is included by the framework. To state one example, voice based recognition of individuals could add valuable information. Since this work focuses on social interactions, it can be expected that speech is a common component.

Last, to allow the results to be compared to recent work in the field as well as measure the degree to which the approach is able to deal with real life data, it would be necessary to extend the dataset. Such an dataset should include recordings of interactions with different individuals, but also the time in between these interactions, as one would expect from non-artificial video material.

Bibliography

[1] Yang Song and Thomas Leung. Context-aided human recognition - clustering. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision - ECCV 2006*, volume 3953 of *Lecture Notes in Computer Science*, pages 382–395. Springer Berlin Heidelberg, 2006.

[2] Padmavathi Mundur, Yong Rao, and Yelena Yesha. Keyframe-based video summarization using delaunay clustering. *International Journal on Digital Libraries*, 6(2):219–232, 2006.

[3] Sanjay K. Kuanar, Rameswar Panda, and Ananda S. Chowdhury. Video key frame extraction through dynamic delaunay clustering with a structural constraint. *J. Vis. Comun. Image Represent.*, 24(7):1212–1227, October 2013.

[4] Silvio Jamil F. Guimarães and Willer Gomes. A static video summarization method based on hierarchical clustering. In *Proceedings of the 15th Iberoamerican Congress Conference on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, CIARP'10, pages 46–54, Berlin, Heidelberg, 2010. Springer-Verlag.

[5] Zheng Lu and K. Grauman. Story-driven summarization for egocentric video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2714–2721, 2013.

[6] Yong Jae Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1346–1353, 2012.

[7] Jean Emmanuel Viallet and Olivier Bernier. Face detection for video summaries. In *Proceedings of the International Conference on Image and Video Retrieval*, CIVR '02, pages 348–355, London, UK, UK, 2002. Springer-Verlag.

[8] Kadir A Peker and Faisal I Bashir. Content-based video summarization using spectral clustering, February 23 2006. US Patent App. 11/361,829.

[9] Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg. Video abstracting. *Communications of the ACM*, 40:55–62, 1997.

[10] S. Lawrence, C.L. Giles, Ah Chung Tsoi, and A.D. Back. Face recognition: a convolutional neural-network approach. *Neural Networks, IEEE Transactions on*, 8(1):98–113, 1997.

[11] Muhammad Ajmal, MuhammadHusnain Ashraf, Muhammad Shakir, Yasir Abbas, and FaizAli Shah. Video summarization: Techniques and classification. In Leonard Bolc, Ryszard Tadeusiewicz, LeszekJ. Chmielewski, and Konrad Wojciechowski, editors, *Computer Vision and Graphics*, volume 7594 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2012.

[12] S. Eickeler, F. Wallhoff, U. Lurgel, and G. Rigoll. Content based indexing of images and video using face detection and recognition methods. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 3, pages 1505–1508 vol.3, 2001.

[13] T. Zhang, D. Wen, and X. Ding. Person-based video summarization and retrieval by tracking and clustering temporal face sequences. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 8664 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, March 2013.

[14] S. Foucher and L. Gagnon. Automatic detection and clustering of actor faces based on spectral clustering techniques. In *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*, pages 113–122, 2007.

[15] Jae-Ho Lee and Whoi-Yul Kim. Video summarization and retrieval system using face recognition and mpeg-7 descriptors. In *Image and Video Retrieval*, volume 3115 of *Lecture Notes in Computer Science*, pages 170–178. Springer Berlin Heidelberg, 2004.

[16] Peng Wang, Yu-Fei Ma, Hong-Jiang Zhang, and Shiqiang Yang. A people similarity based approach to video indexing. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 3, pages III–693–6 vol.3, 2003.

[17] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features, 2001.

[18] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In Josef Bigun and Tomas Gustavsson, editors, *Image Analysis*, volume 2749 of *Lecture Notes in Computer Science*, pages 363–370. Springer Berlin Heidelberg, 2003.

[19] Jean-Yves Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm. Technical report, Intel Corporation Microprocessor Research Labs, 2000.

[20] Jianbo Shi and Carlo Tomasi. Good features to track. In *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593 – 600, 1994.

[21] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 27(5), 2009.

[22] Paul Viola, John C. Platt, and Cha Zhang. Multiple instance boosting for object detection. In *Advances in Neural Information Processing Systems 18*, pages 1417–1426, January 2007.

[23] Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[24] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[25] Yossi Rubner, Carlo Tomasi, and LeonidasJ. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[26] Mohamed Aly, Mario Munich, and Pietro Perona. Multiple Dictionaries for Bag of Words Large Scale Image Search. In *International Conference on Image Processing (ICIP), Brussels, Belgium, September 2011*, September 2011.

[27] Jesper Hojvang Jensen, Daniel PW Ellis, Mads G Christensen, and Soren Holdt Jensen. Evaluation distance measures between gaussian mixture models of mfccs. In *ISMIR 2007: Proceedings of the 8th International Conference on Music Information Retrieval: September 23-27, 2007, Vienna, Austria*, pages 107–108. Austrian Computer Society, 2007.

[28] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.