Efficient People Re-Identification based on Models of Human Clothes

Sebastian Hommel, Darius Malysiak and Uwe Handmann
Computer Science Institute
University of Applied Sciences Ruhr West
Germany, Bottrop

***Abstract*—In this paper, we describe an efficient method for a fast people re-identification based on models of human clothes. An initial model is estimated during people detection and tracking, which will be refined during the re-identification. This stepwise extraction, combination and comparing of features speeds up the whole re-identification. For the refining, several saliency maps are used to extract individual features. These individual features are located separately for any human body part. The body parts are located with an optimized GPU-based HOG detector. Furthermore, we introduce a meanshift-based fusion concept which utilizes multiple detectors in order to increase the detection reliability.**

***Index Terms*—hierarchical people re-identification, clothing model, saliency maps based features, body part detection, nonlinear SVM weights, cluster-based detection, security system, service application**

I. Introduction

The re-identification of people, in various sensor conditions, is a everlasting topic in security applications [1] and will become increasingly important in service applications to recognize interaction partners [2]. To allow a fast re-identification we use a stepwise feature extraction and comparison, combined with a GPU-based body part detection. A fast GPU-based detector is used to enable multiple body part detections at high-resolution images with 10fps. Further fast detection and tracking methods are described in [3], [4], [5].

Several sensors could be used for re-identification, since for a wide range re-identification, cameras are mainly used. A common method is the face recognition [6]. The face recognition allows a re-identification over a long time but it is often not possible to use it with CCTV-Systems[1], since people don't necessarily look into the cameras. Additionally the resolution of a face image can be inadequately low for large distance observations. To allow a people re-identification in such situations, we focus on a model of human clothes for the re-identification of people over a longer period of time (i.e. a few hours). Hahnel [7] compares different methods to describe color and texture of the clothing to model human clothes. A kernel based method to compare simple features of human clothes for re-identification is presented in [8]. Takeuchi shows an improved PCA method for an automatic feature extraction [9]. An online feature selection method for a ranking-based re-identification is shown in [10]. In this work, low level features are selected by comparing several features of current known people, while our method selects conspicuous features separately for each person. Firstly, the full body of each person is detected while the upper body and head areas are separately detected with only one further detector. Secondly, general features are extracted separately at fixed areas at the upper body and the lower body to generate a first clothes model, in contrast to [11] that divides the body shape in regions with a similar appearance. Furthermore, this clothes models are refined with individual features which are located with the help of saliency maps. Other related works which handle similar problems in illumination, pose changing and feature extraction are described in [12], [13], [14].

Before the detection and re-identification process, a camera related dynamic illumination correction which is described in [1] is used.

In this paper, we will present a method for realtime body part detection within high resolution videos. Afterwards, the stepwise extraction of features of human clothes (section III) as well as the stepwise comparison are described. In section V the used testing environment and our results are presented.

II. Body Part Detection

In order to reliably detect people within an image we utilized an algorithm known as *histogram of oriented gradients* (HOG [15]). The algorithms principle can be roughly summarized as follows. It extracts pixelwise edge-gradients from the input image and then assigns each gradient into one of nine orientation bins for a small (e.g. 8x8 pixel) image region. Then, the orientation bins from each image region are sequentially concatenated into a feature vector. This vector is used as the input for a support vector machine (SVM) which is trained for people detection (we will refer to this as a HOG iteration). Once all feature vectors have been binary classified (i.e. once the SVM determined if they may represent an object of interest). The resulting candidates are further reduced through a meanshift algorithm. This algorithm can be applied to a wide variety of objects (i.e. it is not limited to body parts).

A. Nonlinear metric for SVM weights

In order to reduce the computation time and increase the detection quality in the context of HOG applications, all result windows are usually filters out with a SVM weight below

[1]Closed Circuit Television

978-1-4799-5338-7/14/$31.00 ©2014 IEEE

a given threshold t_1, firstly. This strategy can be applied to accommodate the problem of too many items for the mean-shift clustering. Yet, this simple method can also remove a significant amount of correct detections. The reason for this lies in the HOG algorithm. For large objects the corresponding image area is scaled down to the detection window size. This removes a large quantity of high-resolute image information. Such windows will exhibit a smaller SVM weight compared to smaller regions. Filtering according to t_1, which obviously will be chosen according to the higher SVM values (and thus the smaller windows), will remove many candidates for large objects. Our approach addressed this problem under the assumption that not all large windows have been filtered out. We developed a simple strategy by rescaling the SVM weights (after filtering with t_1) according to

$$\omega'(\omega_i) := f(\omega_i)\omega_i \tag{1}$$

with e.g.

$$f(\omega_i) := \begin{cases} \tau \exp(-\beta((I_h - (y_i)_2)/I_h))(y_i)_3 & (y_i)_3 \geq \rho \\ 1 & else \end{cases} \tag{2}$$

with I_h being the image height, ω_i the SVM weight for detection $y_i = (x, y, s)$ ($(y_i)_2$ indicates the second vector component). Each detection y_i contains the detection window center position (x, y) and the corresponding scale s. The general effect of this transformation is that SVM weights of large windows will be increased to rival with those of smaller windows during the mode estimation. This approach not only can retain large windows, it also remove infeasible small windows in an area of large windows. The scaling function f must be chosen to accommodate this goal. In the example above, the scaling function f was chosen for upper-body detections with a camera which covers an area with large depth. Large objects usually appear in the lower part of the image while small objects inhibit the upper portion. Thus SVM weights of window candidates in the lower region should be scaled up, while the scaling should vanish exponentially in the upper image area. Furthermore, only the weights of windows above a certain scale should be transformed, this prevents small windows in the lower area to be transformed as well. An example for this can be seen in Fig.1.

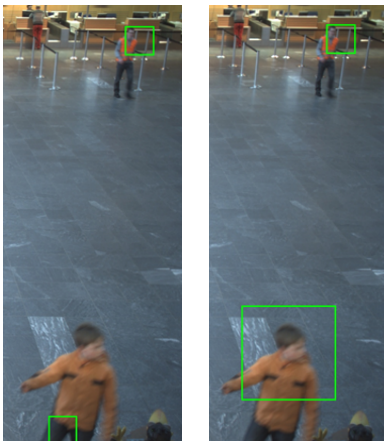

Fig. 1. **Weighting of near field windows** The left image shows the use of a single upper-body detector; the person in the lower part is not detected due to small amounts of candidate windows. Using transformed SVM weights, one can see on the right image that the same detector finds the person in the lower part and suppresses the small false detection now.

B. Cluster-based computation

In the previous section we described how one can increase the reliability of the HOG while keeping the computational costs down. Yet the HOG itself is computationally very expensive. Processing a single frame (1600x1200px) on a CPU can take up to 10 seconds. The computation time for e.g. mean-shifting (30ms-100ms) is completely hidden by that. In order to accommodate this fact we implemented the HOG from scratch in a highly optimized way, reducing its processing time to approximately 60ms for a single detector. Our implementation is mostly system independent as it is written in OpenCL, thus it can be executed on various GPUs as well as CPUs without any change to the source code (the stated times have been determined on a Radeon7970). Such a boost can only be achieved by using modern GPUs and exploit the inherent parallel nature of the HOG.

Running many parallel instances of the HOG algorithm for the same image produces a certain amount of redundant operations. Let us consider the case in which two different objects are being detected within an image, let us further assume that both detectors D_1 and D_2 use the same HOG parameters (i.e. detection window size, stride sizes etc.). In this case the image preprocessing P is identical in both systems, only the SVM classifiers C_1 and C_2 are different. Thus we further optimized our parallel HOG runs (i.e. $(P|C_1)$ etc.) on the same host system by providing a software interface for the described case. The classifiers C_1, C_2 can then work on the same extracted feature set. Our experiments have shown that computation time, in case of two single detectors, can be reduced by up to 37.5%. The detection with two separate HOGs needs 120ms (60ms each), whereby 45ms are required for preprocessing and 15ms for the SVM classification in each single detector. Thus by saving one preprocessing step we save $(1-(45ms+15ms+15ms)/120ms)*100 = 37.5\%$. Currently only the mean-shifting is done on the CPU, thus the largest algorithmic part is being executed on the GPU. During the GPU computation the CPU is completely available for other tasks, thus the CPU can be used for illumination correction before the HOG run as well as extracting first individual features of human clothes.

Initially, we focus to a system structure used to detect all body parts in realtime at high-resolute images (10fps at 1600x1200px) by using only one computer per camera (Fig. 2 left). This approach becomes unfeasible in realistic scenarios by using multiple detectors, as we approach the physical limits of mainstream computers. Thus we developed a software framework to distribute the HOG in a cluster-like manor among small computation nodes, each equipped with one or

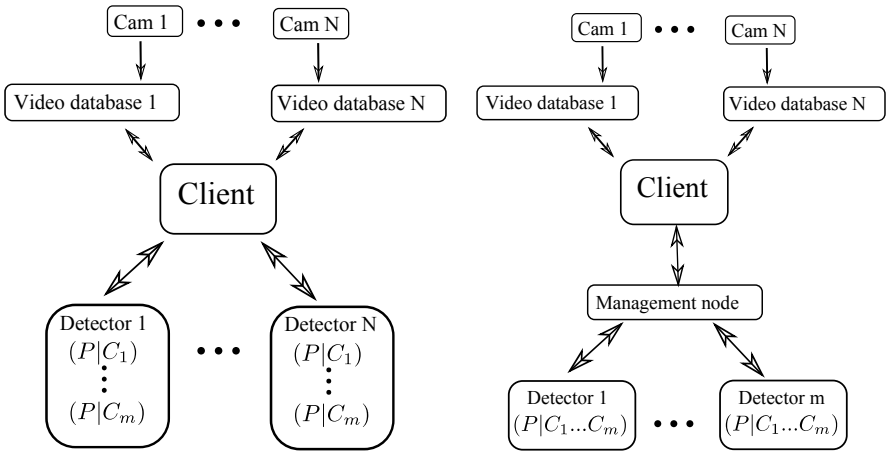

Fig. 2. **Detector system** The left image shows the use of multiple detectors, each executing m iterations $(P|C_i)$. The right image depicts an enchanced version with following differences: a) the workload is distributed via a management node among the detectors b) the detector preprocesses the image only once and uses multiple SVMs

two GPUs. The right side of Fig. 2 visualizes this approach. The systems structure follows the concept of a Beowulf cluster. The systems structure is highly dynamic, new nodes can be added to increase the overall computation power and existing nodes can be removed if the cluster is not fully utilized. Each computation node runs a minimalistic Linux system. During our evaluation, ArchLinux was used although our system can be used on any Linux distribution.

III. Feature Extraction

To speedup the recognition, two types of features are stepwise used. The first kind are the *general features* of human clothes, which are calculated for each person during the detection. These features are used to accelerate the search. In this way, the more complex *saliency maps based features* are only calculated for the searched person and a few hypotheses.

A. General Feature

The used general features are basically described in [2] for a human robot dialog system. Hommel categorizes appearance based features into color and texture features. The texture is naturally independent of the illumination. Whereas the RGB-color representation is transformed into the HSV-representation to use only the illumination independent hue and saturation of the color. The used features are extracted separately at the upper body and lower body (Fig. 3). One rectangle part of the lower body is separated to determine the mean hue and saturation. Furthermore, the mean hue and saturation are calculated at a rectangle part of the upper body, too. At the rectangular upper body part, the mean horizontal and vertical texture rates are calculated with the help of the Scharr filter [16]. The mean horizontal and the mean vertical texture rates describe the strength of the texture at the selected area. One histogram of the hue values and one histogram of the saturation values are calculated at an oval area of the upper body.

Fig. 3. **Feature extraction** The used features for the full body people recognition will be extracted from three areas. This areas are located relative to the whole-body detection.

The mean hue and saturation describes the basic color of the users lower and upper body, while the histograms describes the upper body in detail. By using the hue and saturation histograms, even prints, patches etc. are represented in a very compact manner. To handle minor changes, the hue and the saturation values are divided into 16 parts for the normalized, scale independent histograms. All similar features will be tracked over time, by using the well-known kalman filter, in order to obtain a more robust and faster recognition.

B. Saliency Maps based Features

Additional to the general features we utilize saliency maps in order to describe individual features of the human clothing. To find these features, saliency maps are calculated for each body part. To locate comparable features, it is necessary to warp the body parts to standardized forms. For a faster extraction each body part is warped to a rectangle with a height of 100px. Afterwards, the features will be extracted separately for each body part by a combination of the saliency maps of Itti and Koch [17] with the local entropy at regions of interest [18] (Fig. 4).

This method is described detailed in [1] only for the upper body in combination of RGB- and SWIR-images. In this work four saliency maps are calculated for each body part (upper body, lower body, arms, legs).

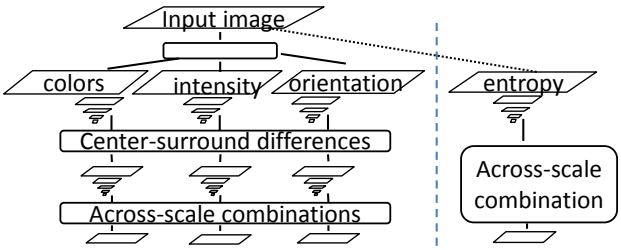

Fig. 4. **Saliency maps** Combination of the saliency maps of Itti and Koch (left) with the local entropy at region of interest (right).

IV. Feature Comparison

To compare the presented clothing features in an efficient manner, firstly the general features between the track of the searched person and each track in the search area are compared. Therefore, the features of each detection of two tracks are compared (Fig. 5). All general features are extracted for each detection during the tracking and detection, to speed up this expensive operation.

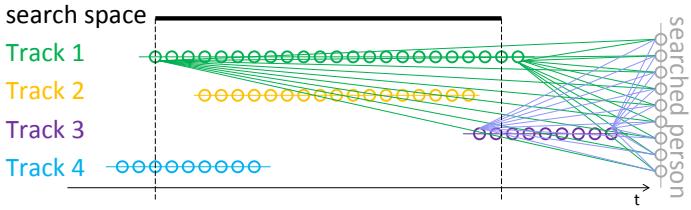

Fig. 5. **Comparing General Features** Each detection of the searched person is compared with each detection of the search area. To display the connections clearly, not each detection is connected in this figure.

During the calculation of the difference D between the hues of the searched person feature space (H_i) and the hue of the current hypotheses (H_c), one must heed that the hue is represented as a circle. In this way, the collection of the errors for each feature is used as a score for each detection. Hence, a small score means a high similarity. The resulted score range between 0 and 1.

For the second step, only the best matching detections between each track of the search area and the searched person is used, once the error is smaller than a threshold (0.06) (see Fig. 6). In this way, the saliency maps based features are only calculated for few detections. To speedup a second search of the same search person or in the same search area, all calculated saliency maps based features are saved, too.

As we wrote before, all the saliency maps based features of each body part of one detection is stored in the matrix M. To recognize a person, the normalized error of the location (normalized euclidean distance) and the related value (normalized absolute difference between the values) of each feature of the matrix M is summarized to a further normalized error. So the saliency maps based error ranged between 0 and 1.

In the next step, the error of the general features and the saliency based features are combined to calculate the final error between two tracks. Firstly, the general feature based

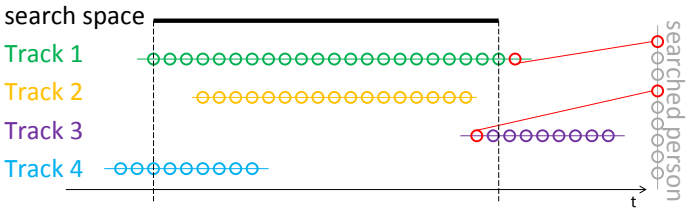

Fig. 6. **Comparing Saliency Maps based Features** The saliency maps based features are only calculated for detections were the error of the first step is smaller than a threshold.

error is divided by the threshold (0.06). So, this error rang for the relevant detections between 0 and 1. Secondly, the saliency maps based error is divided by 2 and the general feature based error is added. At last, the resulted error is divided by 1.5 in order that the resulted error also ranges between 0 and 1.

V. Experiments

We tests our system architecture at an airfield (airport Schönhagen) with private hangars and civil people near the airstrips and at a terminal of a typical general aviation airport (airport Erfurt-Weimar). At the airfield, the CCTV-Cameras are mounted inside and outside, so the re-identification must be able to handle strongly varying illuminations, once the illumination is mostly homogenous inside of one observation area. The illumination at the airport is widely equal between different cameras, since all cameras are mounted in the same arrival hall. The challenge at the airport is the crowding environment. Furthermore, one wall of the hall consists only of windows and glass doors, so there are very dark and bright areas with hard bounds (Fig. 10). At the airfield Schönhagen four cameras are used, two inside a hall (C7 and C6) and two outside (C5 and C4) (Fig. 7).

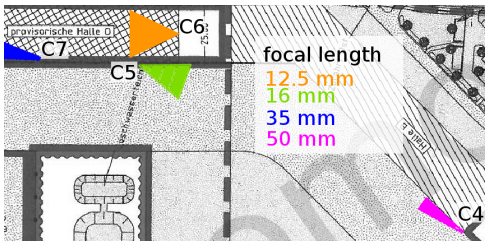

Fig. 7. **Camera position airfield** This figure shows the camera positions at the airfield Schönhagen.

Five cameras are mounted at the airport Erfurt-Weimar (Fig. 8). The cameras C1, C2 and C5 observes the ground floor and the cameras C3 and C4 observes the second floor. At the airfield, three groups of volunteers are recorded. Two groups wear colored casual clothes and one group wear only dark clothes. The figure Fig. 9 shows one image of each camera at the airfield with these three groups.

All the groups walked in several conditions from camera C7 to camera C4 and back. Firstly, the groups walked naturally with small groups inside each of the three groups. Secondly, all groups walked this path loose and afterwards closed. This

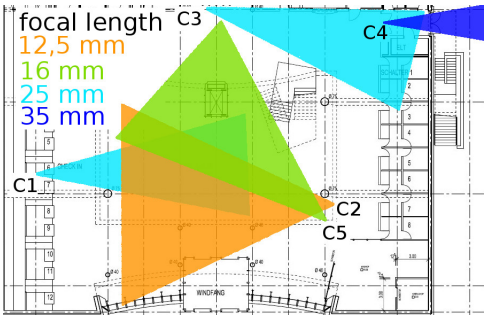

Fig. 8. **Camera position airport** This figure shows the camera positions at the airport Erfurt-Weimar.

Fig. 9. **Camera views at the airfield**

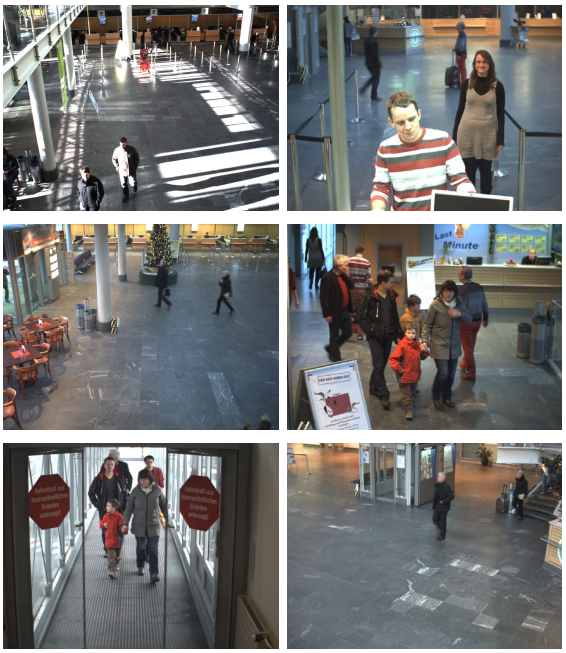

Fig. 10. **Camera views at the airport** The upper left image shows an exemplary image without our illumination correction.

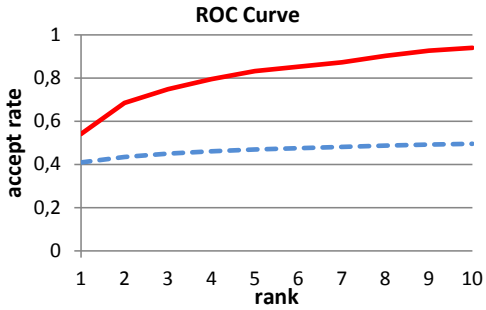

Fig. 11. **Rank 10 statistic** This figure shows the rank 10 statistic of our approach by using the general and the saliency maps based features (red, solid line) as well as without the saliency maps based features (blue, broken line).

will be repeated by splitting each group between camera C6 and C5 outside the observation area. This splitted groups are later merged in the observation area of camera C4.

At the airport Erfurt-Weimar mixed groups are recorded. People with dark business like clothes and people with casual clothes forms up several small, loose groups (Fig. 10). For this scenario the volunteers walked like normal passengers and visitors.

The aim of our method is to ease the people search for human security personal. For that reason, our aim is to rank the searched person over all sequences better than rank 10, which was successfully for 94% of all the fully detected people (Fig 11), by using general and saliency maps based features.

Fig. 12 shows an exemplary result of our re-identification at the airport Schöenhagen and the false acceptance rate (FAR) to each error value for all test sequences at this airport.

A sequence of 10 persons with a length of 4 minutes (2400 frames) can be analyzed in ca. 10 seconds by using a parallel search to use all 4 cores of a Core i7 2,67GHz processor.

VI. Conclusion

In this paper, we presented a method to extract and compare individual clothing features with respect to detected body parts. Two types of features are stepwise extracted. Firstly, general color and texture based features are extracted parallel to the detection process. Secondly, saliency maps based features are extracted only for relevant person detections. The problem of loosing weak correct detection through this process was compensated by a nonlinear transformation of the SVM weights. We showed that a fast and robust re-identification can be realized by tracking body part based clothing features with respect to their similarity in combination with saliency based features (ca. 10 seconds for 24000 person detections; 94% of the recorded people were recognized at range 10 or better). In further work we want to calculate a score for the detections, so the general features will be calculated and compared only for detections with a high certainty. We will evaluate further fast additional methods to locate individual features. Additionally, we will try to stabilize the multi camera re-identification by using additional objects like the camera related tracking in [19]. Furthermore, we want to test our method more systematic with public databases.

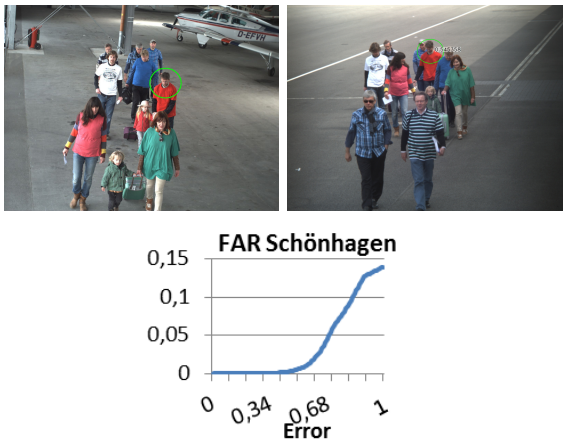

Fig. 12. **Re-identification Schönhagen** The left upper image shows the marked search person at camera C7 at the airport Schönhagen, which is recognized at the camera C4 (upper right), with an error of 0.545358. The lower figure shows that a clear mapping is possible up to an error of 0.6.

References

[1] S. Hommel, D. Malysiak, and U. Handmann, "Model of human clothes based on saliency maps," *CINTI*, pp. 551–556, 2013, budapest, Hungary.

[2] S. Hommel, A. Rabie, and U. Handmann, *Intelligent Systems: Models and Applications*, ser. Topics in Intelligent Engineering and Informatics. Springer Berlin Heidelberg, 2013, vol. 3, ch. Attention and Emotion Based Adaption of Dialog Systems, pp. 215–235.

[3] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 9, pp. 1820–1833, Sept 2011.

[4] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, July 2012.

[5] W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 3258–3265.

[6] U. Handmann, S. Hommel, M. Brauckmann, and M. Dose, *Towards Service Robots for Everyday Environments*, ser. Springer Tracts in Advanced Robotics. Springer Berlin / Heidelberg, 2012, vol. 76, ch. Face Detection and Person Identification on Mobile Platforms, pp. 227–234.

[7] M. Hahnel, D. Klunder, and K.-F. Kraiss, "Color and texture features for person recognition," *IJCNN*, pp. 647–652, 2004.

[8] K. Yoon, D. Harwood, and L. S. Davis, "Appearance-based person recognition using color/path-length profile," *JVCIR*, vol. 17, pp. 605–622, 2006.

[9] Y. Takeuchi, M. Ito, K. Kashihara, and M. Fukumi, "Novel supervised feature extraction algorithm based on iterative calculations," *IRI*, pp. 304–308, 2011.

[10] M. Eisenbach, A. Kolarow, K. Schenk, K. Debes, and H.-M. Gross, "View invariant appearance-based person reidentification using fast online feature selection and score level fusion," *AVSS*, pp. 184–190, 2012.

[11] E. Horbert, K. Rematas, and B. Leibe, "Level-set person segmentation and tracking with multi-region appearance models and top-down shape information," *ICCV*, 2011.

[12] L. Bazzani, M. Cristani, and V. Murino, "Symmetry-driven accumulation of local features for human characterization and re-identification," *Comput. Vis. Image Underst.*, vol. 117, no. 2, pp. 130–144, feb. 2013.

[13] I. Kviatkovsky, A. Adam, and E. Rivlin, "Color invariants for person reidentification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 7, pp. 1622–1634, July 2013.

[14] S. Bak, G. Charpiat, E. Corve, F. Brmond, and M. Thonnat, "Learning to match appearances by correlations in a covariance metric space," in *Computer Vision –ECCV 2012*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Springer Berlin Heidelberg, 2012, vol. 7574, pp. 806–820.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *CVPR*, vol. 1, pp. 886–893, 2005.

[16] H. Scharr, *Optimal operators in digital image processing*. Ph.D. thesis, Interdisciplinary Center for Scientific Computer, Ruprecht-Karls-Universität, Heidelberg, 2000.

[17] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *VISRES*, vol. 40, no. 10-12, pp. 1489–1506, 2000.

[18] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. von Seelen, "Computer vision for driver assistance systems," *Proceedings of SPIE*, vol. vol. 3364, 1998.

[19] T. Baumgartner, D. Mitzel, and B. Leibe, "Tracking people and their objects," *CVPR*, 2013.