Hybrid Person Detection and Tracking in H.264/AVC Video Streams

Philipp Wojaczek[1], Marcus Laumer[1,2], Peter Amon[2], Andreas Hutter[2] and André Kaup[1]
[1]*Multimedia Communications and Signal Processing,*
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
[2]*Imaging and Computer Vision, Siemens Corporate Technology, Munich, Germany*

Keywords: Object Detection, Person Detection, Tracking, Compressed Domain, Pixel Domain, H.264/AVC, Macroblocks, Compression, Color Histogram, Hue, HSV, Segmentation.

Abstract: In this paper we present a new hybrid framework for detecting and tracking persons in surveillance video streams compressed according to the H.264/AVC video coding standard. The framework consists of three stages and operates in both the compressed and the pixel domain of the video. The combination of compressed and pixel domain represents the hybrid character. Its main objective is to significantly reduce the amount of computation required, in particular for frames and image regions with few people present. In its first stage the proposed framework evaluates the header information for each compressed frame in the video sequence, namely the macroblock type information. This results in a coarse binary mask segmenting the frame into foreground and background. Only the foreground regions are processed further in the second stage that searches for persons in the image pixel domain by applying a person detector based on the Implicit Shape Model. The third stage segments each detected person further with a newly developed method that fuses information from the first two stages. This helps obtaining a finer segmentation for calculating a color histogram suitable for tracking the person using the mean shift algorithm. The proposed framework was experimentally evaluated on a publicly available test set. The results demonstrate that the proposed framework reliably separates frames with and without persons such that the computational load is significantly reduced while the detection performance is kept.

1 INTRODUCTION

Over the past few years multiple approaches for video based person detection and tracking have been studied (Yilmaz et al., 2006). Among the most relevant applications is video surveillance of security-relevant areas. This comprises very crowded, open-places like train stations that should be observed to increase the security level. On the other hand there are also security-relevant areas with very few persons present, e.g., in perimeter protection or wide area surveillance applications. In this latter case there is typically a high number of cameras installed. Employing a common object detection and tracking algorithm that evaluates each single frame will waste a lot of resources because most of the frames will not contain any person or the person(s) in the scene will be present only in a small region of the image. To address this problem we propose a new approach to reduce the computational complexity. The basic idea is to create a hybrid framework consisting of three subsequent stages. The first stage will analyze the video streams in the compressed domain, enabling a very fast evaluation of each frame and providing an estimate about the image content. The following stages will only be triggered upon detection of an object in the first stage and will then further evaluate the images in the pixel domain. This way the video decoding and processing is only performed when necessary. The framework's hybrid character results from the combination of compressed domain and pixel domain and the exchange of information between these two domains.

The remainder of the paper is organized as follows: Related work is discussed in Section 2. Section 3 describes the proposed hybrid framework and the algorithms applied. Experimental results of our approach are reported in Section 4, and Section 5 concludes the paper.

2 RELATED WORK

There exists a large number of approaches to fulfill the task of person detection. For instance, among different possibilities for representing objects, like rectangular or elliptical patches or silhouettes, (Yilmaz et al., 2006) present a survey on object detection, seg-

Wojaczek P., Laumer M., Amon P., Hutter A. and Kaup A..
Hybrid Person Detection and Tracking in H.264/AVC Video Streams.
DOI: 10.5220/0005296704780485
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 478-485
ISBN: 978-989-758-089-5

mentation and tracking.

(Eiselein et al., 2013) present a method that improves person detection for crowded scenes based on crowd density measures. They use the histogram of oriented gradients (HoG) for detecting people. As the HoG detector normally provides good results, it fails in the analysis of crowded scenes. Therefore, they create a so-called crowd density map to improve the detection. The crowd density map can be compared to a probability map defining image regions that more or less likely contain people. The map creation is based on feature extraction algorithms like Scale-Invariant Feature Transform (SIFT) (Lowe, 2004). The extracted features are tracked over some frames using Robust Local Optical Flow (RLOF) (Senst et al., 2012) to exclude static features found in the background. The remaining features are weighted with a probability density function (pdf) using a 2-D Gaussian kernel density. The resulting 2-D pdf is the crowd density map, which is used to weight the features found by HoG to improve the detection results.

(Poppe et al., 2009) face the challenge of people detection by establishing a method which evaluates the video sequence in the compressed domain. The idea is based on an observation made on the number of bits in every macroblock. The macroblocks describing the background are well predicted and because of using P frames, the resulting amount of bits is very low. However, when a person enters the scene, the number of bits of the macroblocks containing parts of the person increases, because good compression is difficult to achieve since it is hard to find a reference block. First, the number of bits needed for each macroblock is counted when a frame contains only background. This number of bits serves as a reference. The object detection starts by counting the number of bits of every macroblock in the subsequent frames. If the number of bits of a macroblock increases compared to the number of reference bits of the macroblock, it is likely that this macroblock represents an object.

(Evans et al., 2013) present a multicamera approach for object detection and tracking. The approach follows the idea of projecting a foreground mask onto a coordinate system. The foreground mask is derived from processing each image using the Adaptive Gaussian Mixture Model. The coordinate system is a rectangular grid structure, which is defined on the ground plane of the scene. It is called synergy map. Based on the number of foreground pixels backprojected from each image onto the synergy map, a value is computed which expresses the probability of a present person. An object is created in 3D space based on that value. The object is defined by a 3D bounding box. To track the object in the new frame the dimensions of the 3D bounding box are optimized to fit the object in the new frame. This is done by projecting the 3D bounding box into the images of each camera. The dimensions of the resulting 2D bounding boxes are optimized such that the perimeter of the box surrounds the foreground region. The ideal 3D bounding box is the smallest box, which surrounds all 2D boxes.

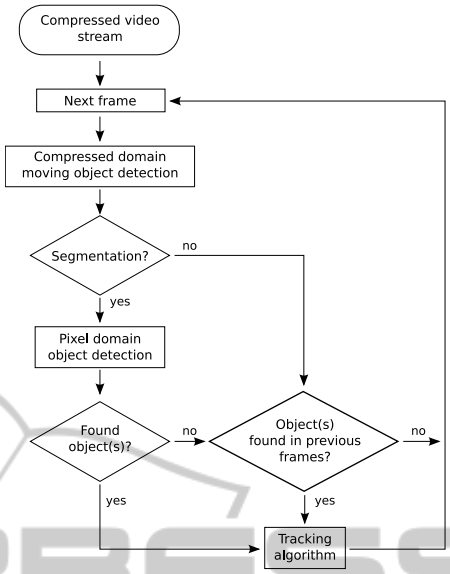

Figure 1: Flowchart of the proposed framework.

There is a drawback in all of the mentioned approaches. The methods analyzing the sequences in the compressed domain are very fast but cannot determine object positions as precisely as pixel domain approaches. On the contrary, methods analyzing the sequences in the pixel domain achieve very good results in people detection but they are generally more complex and therefore often not appropriate for real-time scenarios. Our approach combines both methods. Therefore, it has less complexity but still comparable results in people detection as commonly used pixel domain methods.

3 HYBRID PERSON DETECTION AND TRACKING

Figure 1 shows a flowchart of the framework with the stages “Compressed domain moving object detection”, “Pixel domain object detection” and “Tracking algorithm”. On each stage a different algorithm is applied.

The first stage consists of a compressed domain moving object detection (CDMOD) algorithm. It an-

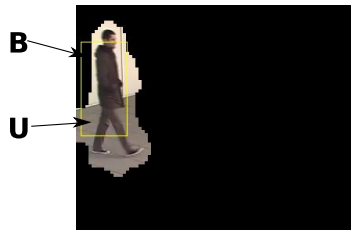

Figure 2: Binary map (black regions) from CDMOD and bounding box (yellow rectangle) from PDOD as overlay on frame 138 from sequence "terrace1-c0".

alyzes each frame of the video sequence which is encoded following the H.264/AVC video coding standard. This algorithm evaluates syntax elements extracted from the bitstream without the necessity of full decoding. Based on that information a binary map for each frame is created. The binary map consists of ones and zeros defining foreground and background of the analyzed image, respectively. Foreground is defined as the image regions, where moving objects are assumed.

If a frame can be segmented into foreground and background, it is likely that there is an object somewhere in the segmented foreground. The binary map is handed over to an algorithm searching for objects in the so-called pixel domain: the pixel domain object detection (PDOD). Therefore, the frame has to be decoded before analysis. The second algorithm is needed, as the CDMOD can not differentiate between objects of different types. For example, slightly moving trees or noise would also cause a binary mask. It just provides a binary mask with a coarse segmentation. The PDOD algorithm analyzes the image of the sequence only in the foreground defined by the segmentation made by the CDMOD. If no object can be detected, the next frame will be analyzed by the CDMOD. Otherwise, if an object is found, information describing the object's position and scale are handed over to the last stage, the tracking.

It is most likely that the PDOD detects the same object again in consecutive frames. Therefore, new detections are regarded as candidates for new, still unknown persons and the tracking algorithm initially tries to match new detections with already known objects. It should be guaranteed that only as many objects are tracked as actually available in the sequence. Each detected object is tracked in following frames. As soon as the tracking algorithm's processing comes to an end the next step is CDMOD analyzing the next frame of the sequence. Details to the employed algorithms are given in the following subsections.

A person that does not move in consecutive frames, remains undetected by the PDOD since the CDMOD detects only moving objects and does not create a binary mask. But the person had to move to appear in the image plane. Therefore, a binary mask is created as soon as it enters the scene and it is very likely that the person is detected before stopping. The person's position is known and it can be tracked. In case the person stops moving, the position remains constant until it moves and can be tracked again.

3.1 Compressed Domain Moving Object Detection

The approach described in (Laumer et al., 2013) is used as CDMOD algorithm. The algorithm analyzes the type of macroblocks used for compression of videos in the H.264/AVC video coding standard (MPEG, 2010). The available macroblock types are grouped to so-called macroblock type categories (MTC). Each of the MTC gets a specific weight, the macroblock type weight (MTW). The higher the weight the higher is the assumed probability of the presence of motion in that macroblock. By analyzing the macroblocks of each frame a map consisting of MTWs can be created. As the area of moving objects spans over multiple macroblocks, the map is further processed such that each macroblock weights its neighboring macroblocks according to its own weight. The resulting map is thresholded. The resulting binary map defines foreground and background. An example of a binary mask can be seen in Figure 2. A prediction can only be made for P frames. In the case of an I frame only Intra-Frame prediction is allowed. In our configuration we use the binary mask from the previous P frame again as binary mask for the I Frame.

3.2 Pixel Domain Object Detection

To find actual objects within segmented frames, the Implicit Shape Model (ISM) is used (Leibe et al., 2004). The method consists of a training and a detection phase. The training phase is done offline. Training images are searched for keypoints. A codebook is created, based on computed descriptors describing the keypoints. In the detection phase, the objects trained on are searched in the image. An image is searched for keypoints and the descriptors are matched to the codebook. The higher the match, the more likely a person is detected.

In our configuration the feature detection is done with the SIFT algorithm (Lowe, 2004). Unlike the approach described in (Leibe et al., 2004), in our configuration a final segmentation based on pixels is not used. Our configuration of the ISM describes the position of found objects with four parameters: x, y, w,

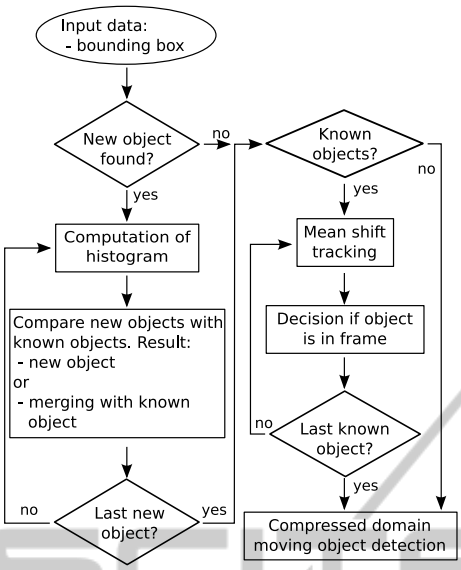

Figure 3: Flowchart to the tracking algorithm.

and h. The parameters describe the coordinates of the upper left corner and the width and height of a rectangle surrounding a found object. For an example see Figure 2, where a bounding box is visualized as a yellow rectangle.

The training is based on the TUD-Pedestrians training data set (Andriluka et al., 2008). It consists of 400 images showing several persons. Each image is scaled such that each person has the equal height of 200 pixels.

3.3 Tracking Algorithm

Figure 3 shows the flowchart of the tracking algorithm. For each found object four parameters are provided as input as described in Section 3.2. A color histogram for each object is computed. This histogram is used for the matching of new objects with known objects and for tracking. Based on that matching, either an unknown object is found or it is merged with a known object. Finally, new positions for all known objects are evaluated based on the mean shift algorithm (Comaniciu et al., 2003). The mean shift algorithm shifts the bounding box from the known object's position to a new one. As the mean shift algorithm needs a probability density function (pdf) describing the object to track, the previously computed color histogram is reused. The next step is the decision if the object is leaving the frame. Finally, the next frame is analyzed by the CDMOD.

3.3.1 Computation of Histogram

If an object, i.e. a person, is found, a color histogram as mentioned above needs to be computed. The bounding box usually contains too much background since it is too coarse for describing people's complex geometries. And as neither the person's color appearance is known in advance nor the person's geometry is rigid, the following method to segment the person automatically with less background is established: the tracking algorithm receives information, on the one hand the bounding box describing the person's position and on the other hand the binary map. These two descriptions are combined as shown in Figure 2.

The image's black part is background declared by the CDMOD. The yellow rectangle is the visualized bounding box from the PDOD. It can clearly be seen that a rectangle does not fit a person's shape very well, as the person's geometry is too complex. Some parts of the background lie inside the bounding box, which is marked with **B**. The remaining regions inside the bounding box are defined as **U**. It is still unknown which part of **U** belongs to foreground or background. In the next step of the algorithm, a histogram $\mathbf{h}_B$ for **B** is computed. We selected the hue component from the color space HSV for histogram computation, because (Corrales et al., 2009) stated good results in segmenting objects with hue. After evaluating the histogram $\mathbf{h}_B$, histograms $\mathbf{h}_i$ for smaller image regions belonging to **U** are computed. Image regions defined as **B** are not taken into account. Every smaller image region $U^i_{N\times N}$ is of size $N \times N$ pixel. In our configuration N is set to 4. Additionally, for every histogram $\mathbf{h}_i$ the Bhattacharyya distance d_i to $\mathbf{h}_B$ is calculated according to (Kailath, 1967). The next step is the segmentation of **U** into foreground and background. Every block $U^i_{N\times N}$ is compared to the background **B** by comparing its distance d_i to a threshold t_{Avg}. t_{Avg} is based on the average distance $d_v = \frac{1}{K}\sum_i^K d_i$, where K is the number of blocks inside **U**. We set $t_{\text{Avg}} = 1.2d_v$. If $d_i < t_{\text{Avg}}$ then $U^i_{N\times N}$ has more in common with the background than a fictive average block of size $N \times N$ inside **U** and $U^i_{N\times N}$ can be labeled as background. Otherwise, if $d_i > t_{\text{Avg}}$, then the block $U^i_{N\times N}$ is less similar to the background than the average block and it is labeled as foreground. Based on the labeling with foreground and background, a binary map similar to the one from the CDMOD can be created. Based on the map, a color histogram in hue describing the person's appearance is calculated. An example of a segmentation is shown in Figure 4. The person is well segmented but some parts of the object's head and feet are labeled as background. The reason is that the bounding box does not surround the person totally, as the yellow bounding box in Figure 4 shows. Only the inner parts of the bounding box are taken into account from the above described algorithm, therefore, these object's parts are

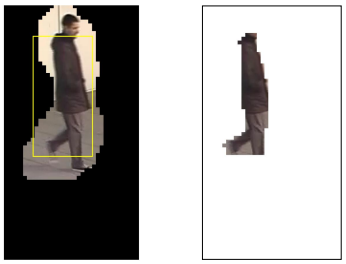

Figure 4: Example of a segmentation of a person based on hue. Cutouts from Frame 145 from sequence "terrace1-c0". (a) black regions: binary mask; yellow rectangle: visualization of bounding box. (b) resulting segmentation of the object.

missing.

3.3.2 Matching of Objects

A newly found object by the PDOD is called a candidate. It can either be a new, unknown object or a person whose presence is yet known. The decision whether the candidate is a yet known person is based on two criteria calculated by the tracking algorithm: the amount of overlap of the bounding boxes and the matching of both histograms. The overlap O of two bounding boxes is calculated according to:

$$O = \frac{A_n \cup A_k}{A_n \cap A_k}, \tag{1}$$

where A_n and A_k are the areas from the rectangles defining the bounding boxes from the newly found and the known objects respectively. The matching of the histograms $\mathbf{p}$ and $\mathbf{q}$ from the known and the newly found objects is done according to:

$$d = \sqrt{1 - \rho[\mathbf{p}, \mathbf{q}]}, \tag{2}$$

where $\rho[\mathbf{p}, \mathbf{q}] \equiv \sum_{u=1}^{m} \sqrt{p_u q_u}$ is the so-called Bhattacharyya coefficient (Kailath, 1967). (2) measures the distance between two color histograms and is bounded to $[0, \ldots, 1]$. The smaller d, the higher is the similarity between two color histograms. (1) and (2) are calculated between a new object and every known object and are compared to two individually chosen thresholds t_O and t_H:

$$1 - O \leq t_O \tag{3}$$

$$d_i < t_H \tag{4}$$

If the candidate and a known object fulfill (3) and (4) they are merged, since it is assumed that several detections represent the same person but are detected a multiple times by the PDOD.

3.3.3 Using the Binary Mask for Tracking

As it is possible that the mean shift algorithm converges to false positions due to similarities between the background and the persons appearance, the binary mask is used to force the mean shift algorithm to converge only in the defined foreground. The bounding box is shifted from the old object's position to a new one. The new position is in the segmented foreground. Even if the background is similar to the object and converging to the background is likely, the bounding box is still near the actual object's position.

3.3.4 The Bounding Box as Binary Mask

To further enhance the tracking results and to reduce the amount of computations, the bounding box from each known object is used in addition to the binary mask. That means as soon as an object is detected, its bounding box is also used as binary mask in the following frames and defined as image region, in which feature points should not be searched. Feature points can not be detected in the bounding box's region and therefore the object is not detected a second time. This is useful as it is not necessary to detect a known object again.

4 EVALUATION

The framework has been evaluated with video sequences from a data set of CVLAB (Berclaz et al., 2011). All video sequences have been encoded using the H.264/AVC Baseline profile with a GOP size of ten frames.

4.1 Measures for Video Analysis

As evaluation measurements we used two different criteria. For the segmentation of objects we used precision and recall, defined as:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

where TP defines the true-positives, the number of correct detections, FN (false-negatives) is the number of missed detections and FP (false-positives) the number of false detections.

The outcome of our framework is not pixel-based but each found object is described with a bounding box. It is not appropriate to use again the measurements recall and precision because a parameter defining the minimal amount of overlap has to be chosen

Table 1: Results for object segmentation based on hue.

Sequence	Recall	Precision
campus4-c0	0.33	0.76
campus7-c1	0.51	0.79
terrace1-c0	0.40	0.77
terrace2-c1	0.25	0.66

Table 2: METE for using only the PDOD algorithm and for CDMOD and PDOD algorithm.

Sequence	METE for	
	PDOD	CDMOD & PDOD
campus4-c0	0.71	0.76
campus7-c1	0.91	0.38
terrace1-c0	0.78	0.61
terrace2-c1	0.74	0.57

to count as true-positive. Such a hard decision of a threshold makes is difficult to compare the tracking results. The Multiple Extended-target Tracking Error (METE) described in (Nawaz et al., 2014) is independent of such parameters. Therefore, we chose METE to evaluate the performance of the framework. First, the accuracy error $\mathcal{A}_k$ and the cardinality $\mathcal{C}_k$ for each frame k are calculated. $\mathcal{A}_k$ represents the accuracy error in frame k: $\mathcal{A}_k = \sum_i \mathcal{A}_k^{ij}$, where $\mathcal{A}_k^{ij}$ defines the amount of overlap between the area of a bounding box of a tracked object i and the area of a bounding box of a ground-truth object j. $\mathcal{C}_k$ is the difference between estimated objects u_k and ground-truth objects v_k. METE is calculated as:

$$\text{METE}_k = \frac{\mathcal{A}_k + \mathcal{C}_k}{\max(u_k,\ v_k)} \tag{7}$$

METE_k is bounded to $[0, \ldots, 1]$, where zero is the best result.

4.2 Evaluation Results

First, we evaluated the results of the object segmentation based on the hue component, please see Section 3.3.1. Table 1 shows the results of the segmentation.

The values for recall are quite low compared to the precision values. This is mostly because the bounding boxes from PDOD do not surround the objects contour in total but exclude some parts of the body, like the feet, head or even the complete upper part of the body. The excluded parts are not taken into account for segmentation into foreground and background, which leads to low recall values. Some objects in the video sequences are wearing clothes which are partially similar to the background. They are also often labeled as background, which lowers the recall values as well.

For evaluation of the framework we compared the performance of the framework to the performance of the algorithms when used separately. First we used only the PDOD algorithm without the binary mask from the CDMOD and the tracking algorithm. That means every image from a sequence is searched for objects without separating image regions into foreground and background. The results for using only the PDOD algorithm can be seen in Figure 5 exemplarily for sequence terrace1-c0. METE is shown per frame. That means the detection results are compared to the ground truth of each frame. The error is influenced from missed detections and false detections. True detections where the overlap of bounding boxes is not accurate enough also influence the error.

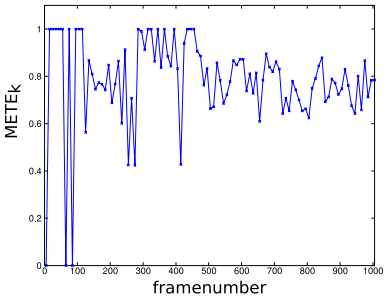

Figure 5: Error METE_k per frame k for sequence *terrace1-c0* using PDOD algorithm only.

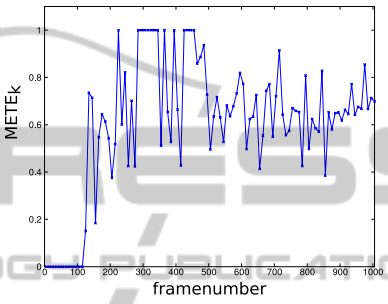

Figure 6: Error METE_k per frame k for sequence *terrace1-c0* using the concatenation of CDMOD and PDOD algorithms.

Especially at the beginning of the sequence the error METE equals 1. In this sequence there are no persons which can be detected in the first 100 frames. That means the error cannot result from missed detections or detections with insufficient overlap but from false-positive detections. This shows the influence of false-positive detections on a tracking system. In the next frames the error is influenced from false and missed detections and from insufficient overlap of bounding boxes. Figure 6 shows the results when using the combination of CDMOD and PDOD algorithm exemplarily for sequence terrace1-c0.

The comparison of Figure 5 and 6 shows that METE equals 0 at the beginning of the sequence for

Table 3: Time of analysis for 25 frames[s] for PDOD only and PDOD with CDMOD.

Sequence	PDOD	PDOD & CDMOD
campus4-c0	2.9	2.5
campus7-c1	2.9	1.5
terrace1-c0	3.6	3.1
terrace2-c1	4.1	3.7

Table 4: Time of analysis for 25 frames[s] for the framework.

Sequence	Framework	
	w/o extensions	with extensions
campus4-c0	1.1	1.0
campus7-c1	0.8	0.8
terrace1-c0	1.4	1.2
terrace2-c1	1.7	1.3

the first 100 frames. The CDMOD algorithm creates a binary mask for each frame. As mentioned before, there are no moving objects in the first 100 frames of that sequence. Therefore, the binary mask defines each frame of the first 100 frames as background and the PDOD algorithm does not search for objects. This shows the advantage of the CDMOD algorithm. It separates frames without persons from frames with persons and prevents false-positive detections which would be tracked wrongly.

The averaged error METE is listed in Table 2. In the second and third columns are the results for PDOD and the combination from CDMOD and PDOD listed. As expected, the error decreases, only for the sequence campus4-c0 it increases. This is because of some persons are not moving in some frames, so they are declared as background and can not be found from the PDOD algorithm.

As described in Section 3.3.2 the matching of objects depends on two parameters, on the one hand the amount of overlap of the bounding boxes and on the other hand on the similarity of the color histograms. In the evaluation we parameterized over two thresholds t_o and t_s to achieve the best overall values which should be used for object merging. Both thresholds define the maximal value of the overlap o_{ij} and the similarity of histograms s_{ij} of a new detected object i and a previously found object j may reach. If $o_{ij} < t_o$ and $s_{ij} < t_s$ both objects are merged. Additionally the PDOD algorithm analyzed every 5th frame only, because once an object is detected it is not necessary to detect it again in the following frame, but the CDMOD and the tracking algorithm are still analyzing each frame. Another benefit is the reduction of time.

The computation time for the analysis with PDOD only and for PDOD and CDMOD is listed in Table 3. As expected, the time of computation could be reduced for using the combination of the CDMOD and PDOD algorithms compared to using only the PDOD algorithm. The CDMOD algorithm defines background in images, which is not searched for objects from the PDOD algorithm. This leads to the reduction of computation time. In sequence campus7-c1 only one person appears in approximately 50% of the sequence. The computation time is reduced by almost 50%. This shows that the CDMOD algorithm is very well suited to reduce the complexity. Table 4 shows the computation time when the analysis is done with the framework, that means the combination of CMDOD, PDOD and the tracking algorithm. The computation time is again reduced to approximately 1 second for 25 frames. This result shows that the maximum computation effort is made from the PDOD algorithm, as the PDOD algorithm analyzes only every 5th frame in this configuration.

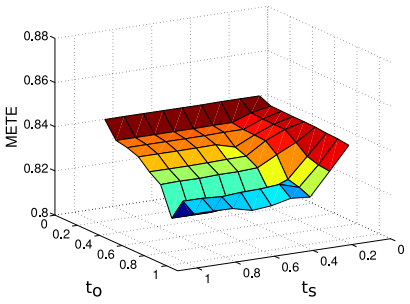

Figure 7: Error METE averaged for sequence *terrace1-c0* using the framework.

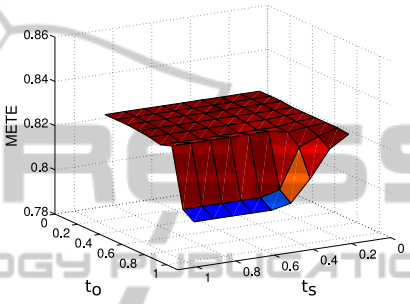

Figure 8: Error METE averaged for sequence *terrace1-c0* using the framework and the extensions.

In Figure 7 the averaged METE is shown when parameterizing both thresholds. METE is not shown per frame but averaged for the sequence with fixed parameters. In Figure 8 the averaged METE is shown for using the extensions described in Section 3.3.3 and Section 3.3.4.

Unfortunately, the error increases compared to the results in Table 2. The main reason is that there is no knowledge about previous true-positive or false-positive detections, when using the PDOD algorithm only or combined with the CDMOD algorithm. On

the contrary when using the framework, each false-positive detection is tracked in the subsequent frames. That means if an object is detected repeatedly but a matching was not successfully, the object is tracked with more than one bounding boxes. Even if all bounding boxes follow the object correctly the cardinality error increases, which results in a high METE. But as one can see, a low error is reached for a low threshold t_s. But the threshold t_o has to have a high value to achieve a low METE. That means the color histogram is more suitable for object merging, than the overlap of bounding boxes. Another influence on the error is the mutual occlusion of objects. The mean shift algorithm is not able to follow a hidden object, instead it converges to false positions.

5 CONCLUSION

In this paper we presented a framework for the detection and tracking of objects. The framework consists of three stages. For each stage an individual algorithm is applied. The stages are concatenated in a way that they exchange information about the presence and the position of objects. An algorithm analyzing the compressed video stream is used as a preselection step to provide a binary mask, which segments the regions of an image into foreground and background. We selected the Implicit Shape Model as algorithm to actually find the position of objects. A tracking algorithm using the mean shift algorithm was established to track the detected objects. The novelties lie in the concatenation of algorithms analyzing the video sequence in the compressed domain and pixel domain. Another novelty is the method of object segmentation to receive a color histogram, as it is needed for the mean shift algorithm. The evaluation results state good results in object segmentation and tracking when using the new method. It is also shown that the complexity could be reduced significantly. Another challenge is multiple person tracking and mutual occlusion of persons. This could be handled with previous knowledge like evaluation of the individual trajectory for example.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from EIT ICT Labs' Action Line "Future Cloud" under activity n^{o} 11882.

REFERENCES

Andriluka, M., Roth, S., and Schiele, B. (2008). People-Tracking-by-Detection and People-Detection-by-Tracking. In *Proc. 2008 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

Berclaz, J., Fleuret, F., Turetken, E., and Fua, P. (2011). Multiple Object Tracking using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819.

Comaniciu, D., Ramesh, V., and Meer, P. (2003). Kernel-based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577.

Corrales, J., Gil, P., Candelas, F., and Torres, F. (2009). Tracking based on Hue-Saturation Features with a Miniaturized Active Vision System. In *Proc. 40th Int. Symposium on Robotics*, pages 107–112.

Eiselein, V., Fradi, H., Keller, I., Sikora, T., and Dugelay, J.-L. (2013). Enhancing Human Detection Using Crowd Density Measures and an Adaptive Correction Filter. In *Proc. 2013 10th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS)*, pages 19–24.

Evans, M., Osborne, C., and Ferryman, J. (2013). Multicamera Object Detection and Tracking with Object Size Estimation. In *Proc. 2013 10th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS)*, pages 177–182.

Kailath, T. (1967). The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Transactions on Communication Technology*, 15(1):52–60.

Laumer, M., Amon, P., Hutter, A., and Kaup, A. (2013). Compressed Domain Moving Object Detection Based on H.264/AVC Macroblock Types. In *Proc. of the International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 219–228.

Leibe, B., Leonardis, A., and Bernt, S. (2004). Combined Object Categorization and Segmentation With an Implicit Shape Model. In *Proc. Workshop on Statistical Learning in Computer Vision (ECCV workshop)*, pages 17–32.

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

MPEG (2010). ISO/IEC 14496-10:2010 - Coding of Audio-Visual Objects - Part 10: Advanced Video Coding.

Nawaz, T., Poiesi, F., and Cavallaro, A. (2014). Measures of Effective Video Tracking. *IEEE Transactions on Image Processing*, 23(1):376–388.

Poppe, C., De Bruyne, S., Paridaens, T., Lambert, P., and Van de Walle, R. (2009). Moving Object Detection in the H.264/AVC Compressed Domain for Video Surveillance Applications. *Journal of Visual Communication and Image Representation*, 20(6):428–437.

Senst, T., Eiselein, V., and Sikora, T. (2012). Robust Local Optical Flow for Feature Tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(9):1377–1387.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object Tracking: A Survey. *ACM Computing Surveys*, 38(4).