2009 Advanced Video and Signal Based Surveillance

Multi-Cue Based Visual Tracking in Clutter Scenes with Occlusions

Jie Yu, Dirk Farin, Hartmut S. Loos
Corporate Research Advance Engineering Multimedia
Robert Bosch GmbH
Hildesheim, Germany
{*jie.yu, dirk.farin, hartmut.loos*}*@de.bosch.com*

***Abstract*—Object tracking is important for video analysis applications. However, tracking through occlusions is a difficult task due to significant appearance changes of the objects. Approaches based on either global features or one kind of local features can not solve the problem completely. In this paper, a multi-cue based tracking approach is introduced. It combines a corner tracking with a color and a shape model to resolve the object tracking problem through occlusions for most scenes (indoor and outdoor). To obtain an objective evaluation of the proposed method, a set of detection and tracking measures are used to perform a quantitative analysis based on a large sequence dataset with ground-truth annotation. The experimental results show that the proposed approach works robustly under varying conditions.**

I. Introduction

Object tracking is a key issue in different video-based applications, such as visual surveillance, video archival and retrieval systems, robotics, *etc.* . The results of tracking can be used to improve the performance of object recognition, object classification and high-level event understanding. However, tracking through occlusions is still an open problem. The main difficulty in tracking multiple objects with occlusions is to maintain the object model while the object appearance changes significantly. To handle this problem, corner features are a good candidate, as they are easy to compute and well localized. They keep their relative position on the object even when the tracked object shows significant changes of shape or if it becomes partly invisible. Both are typical situations during occlusion. In the literature, there are several approaches based on corner features. In Asadi *et al.* [1], and Regazzoni and Asadi [11], corners combined with relative spatial features are proposed to localize the object center using a voting space. In order to be robust against background clutter, the tracked corners are classified based on voting results and only good corners are used to update the object model. A corner-based method that uses wavelet features to generate descriptors is proposed in Asadi and Regazzoni [2]. In Zhu *et al.* [15], local image patches are extracted. Color information and relative spatial positions of these patches are used to train classifiers which are adapted online during the tracking. In Kim [8], the extracted corner features are tracked in two successive frames by template matching and then clustered based on their positions, motions and membership history.

The drawback of those approaches is that they depend highly on the texture features of the objects and do not work well if objects have relatively large homogeneous regions. As compensation, other object features are considered. Sidla *et al.* [12] propose a pedestrian-tracking algorithm based on KLT (Tommasini *et al.* [13]) tracking and shape matching. However, calibration has to be given manually due to the fact that contour features are not scale invariant. Besides, they are also sensitive to occlusion. Another important feature that is widely applied to model object appearances in vision tasks is the color distribution. It is robust to noise and suitable to model objects with partial occlusion. In [5], [6], [14], a color distribution is used to build an object model. They employ a gradient optimization method for the target localization. An object model based on color histograms and the spatial distribution of features is proposed in Huang and Essa [7] to localize objects in an occlusion situation.

The main contribution of this paper is to introduce a new framework for tracking through occlusions. Corner features, spatial features, and color information are integrated into a probabilistic framework. By employing color features, a global view of the object is obtained and it can be updated online even when occlusion occurs. Based on our approach, a complete visual tracking system is built and evaluated with real video sequences by comparing to ground truth data. It is shown that the proposed method improves the performance of the individual features greatly.

This paper is organized as follows: Section 2 gives a description of our system. In Section 3, experimental results are shown, and we conclude in Section 4.

II. System Description

The goal of our system is to track multiple objects in cluttered scenes, such that their identities are maintained as long as possible. In order to detect multiple objects, a background model is used. A background-subtraction algorithm is employed to extract regions of interest in the form of foreground-object masks in which each connected component is considered a blob. Note that each blob can consist of more than one real-world object. In the beginning, each blob is considered a single real-world object. Object blobs are tracked by considering the overlaps between blobs in successive frames. If a blob is splitted into two objects,

978-0-7695-3718-4/09 $25.00 © 2009 IEEE
DOI 10.1109/AVSS.2009.56

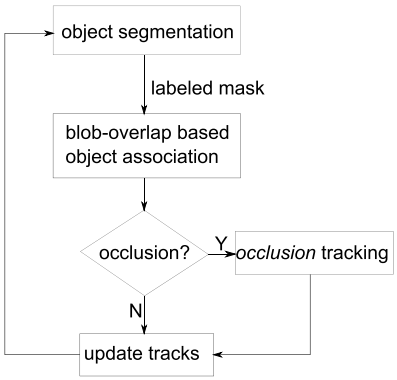


Figure 1. Overview of the proposed tracking system.

both are tracked independently. When two blobs merge into the same successor-blob, the tracking algorithm described in the following sections is employed instead of the simple overlap rule. Instead of merging the two blobs into a single object, our tracking algorithm keeps the two blobs separate, so that both objects can be tracked separately through the occlusion. An overview of the tracking system is shown in Figure 1.

A. Object Segmentation

The background is modeled as a static background image B. To handle the difficulties of illumination changes and background content changes, an adaptive background update is introduced [9]. By subtracting the background from the image I_t at the current frame t, a difference image is obtained. It is thresholded with γ to obtain a binary change mask, i.e. the foreground-object mask:

$$M^t = \{\mathbf{x} \mid |I_t(\mathbf{x}) - B(\mathbf{x})| > \gamma\} \quad (1)$$

This mask is separated into disjoint object masks M_i^t with a connected-component algorithm.

B. Occlusion Detection

The association of object i in the current frame with object k in the previous frame is established if their areas overlap $M_i^t \cap M_k^{t-1} \neq \emptyset$. If two objects i_1 and i_2 are associated with the same object k, a merging of objects is detected and a special tracking process will be started to decompose the merged objects and to adjust the association of objects across frames. This special tracking process will be described in the following sections.

C. Object Representation

In order to handle the occlusion situation, the merged objects should be represented in a proper form so that an accurate localization in a new frame is possible. A combination of bottom-up and top-down model is used, where the former is based on a set of corner points and the latter maintains a color distribution for each object.

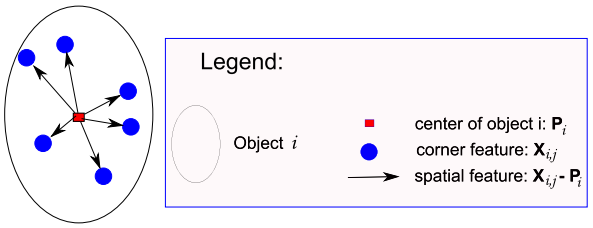


Figure 2. The object representation based on the corner points.

1) Corner Model: To deal with appearance changes during occlusions, a bottom-up description of the objects is used. To this end, a set of corners $\{\mathbf{X}_{i,1}^t, \mathbf{X}_{i,2}^t, \ldots, \mathbf{X}_{i,m}^t\} = \{(x_{i,1}^t, y_{i,1}^t), (x_{i,2}^t, y_{i,2}^t), \ldots, (x_{i,m}^t, y_{i,m}^t)\}$ inside of the object mask M_i^t is extracted and considered as object features, where the pair $(x_{i,j}^t, y_{i,j}^t)$ is the absolute coordinate of the j-th corner feature of object i. In addition, spatial features for each corner are computed like in Asadi *et al.* [1] to encode the shape information of objects. The center $\mathbf{P}_i^t$ of the i-th object is chosen as a reference point and the spatial features are computed relative to the reference point: $\mathbf{S}_i^t = \{\mathbf{X}_{i,1}^t - \mathbf{P}_i^t, \mathbf{X}_{i,2}^t - \mathbf{P}_i^t, \ldots, \mathbf{X}_{i,m}^t - \mathbf{P}_i^t\}$. This is depicted in Figure 2. The advantage of this model is that it includes shape and position of the objects and partial occlusion has minor influence on the entire model. Some features may be occluded, but the relative position of the feature to the object center stays stable. However, the corner model also has some disadvantages: the geometric relation between features inside of the object is weak and it is assumed that there is always sufficient object texture for the extraction of corner features. The drawbacks could make the tracking unstable in situations with low texture and ambiguities in the feature locations.

2) Color Model: To overcome the shortcomings of the corner model, a top-down model is needed. An example of a simple and good global feature for object tracking is the color histogram. It describes the global color distribution, is robust against partial occlusion and gives a good summarization of an entire object. Another important point is that no texture is required for the computation. These two points make it a good, complementary extension to the corner model. However, color histograms ignore the spatial relation or layout of the colors, which can lead to confusion when the background has similar colors. In the next section it will be discussed how a color histogram can be combined with the corner model to track occluded objects in cluttered scenes.

Since the peripheral pixels are the least reliable, being often affected by occlusions (clutter) or interference from the background, a kernel function is used to assign large weights to pixels near the object center and smaller weights to peripheral pixels. This increases the robustness of the estimation of the color histogram. More specifically, let

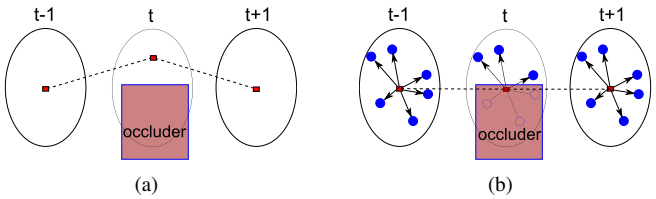


Figure 3. Object localization using (a) color and (b) corner models. For the color model, the object center is derived from the visible object mask. It leads to a bias during occlusions. On the other hand, the corner model defines the object center relative to the detected features. Even if some features are occluded, the object center stays stable.

$\{\mathbf{x}_i\}_{i=1\ldots n}$ be the pixel positions normalized to the range $x, y \in [-1, 1]$ within the object bounding box. The probability of the color u in the target is then computed as

$$\mathbf{CD}(u) = \frac{1}{C}\sum_{i=1}^{n} K(\mathbf{x}_i)\delta[b(\mathbf{x}_i) - u], \tag{2}$$

where δ is the Kronecker delta function, C is a normalization constant, $b(x_i)$ computes the quantized color at position x_i, and $K(\cdot)$ is the kernel function. In our experiment, the Epanechnikov kernel is used:

$$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|\mathbf{x}\|^2) & \|\mathbf{x}\| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where c_d is the volume of the d-dimensional unit sphere.

3) Combination of corner and color models: The corner and color models have both specific advantages and disadvantages. However, the combination of both models can lead to a superior result compared to employing each model alone.

For example, the color model usually gives a complete object mask, while the corner features are only located in textured areas. On the other hand, the color model can fail if the foreground color is similar to the background color. In this case, the corner model can help to complete the object mask, as the corner tracking is independent of color.

Furthermore, the combination of features can improve the object localization. As the color model does not explicitly provide localization information, it has to be derived indirectly, e.g., as the center of the obtained object mask. This can lead to a bias in the derived object position during occlusions (see Figure 3a). On the other hand, the corner model provides a direct connection between the object position and each corner feature. The estimation of the object position is not influenced, even when some corner features are invisible or lost. Hence, object localization stays robust through partial occlusions (see Figure 3b).

The focus of the following section is to model the joint probability of an assignment of each pixel to a particular tracked object.

D. Probabilistic Tracking Model

The proposed tracking method consists of two components: a corner tracking and a color-model. Considering our special tracking situation, i.e., two or more objects participating in an occlusion and being merged in the foreground mask from the background subtraction, the problem can be reduced to a labeling problem. Let $L = \{l_1, l_2, \ldots, l_n\}$ be the label set for all tracked objects in the occlusion. The task is to estimate the posterior probability of each pixel within occluded objects in the current frame to belong to a particular object. This is expressed as the probability $p(L|\mathbf{X}, \mathbf{Z})$ of assigning a particular label L, where $\mathbf{X}$ is the pixel coordinate and $\mathbf{Z}$ is the observation at the pixel. With Bayesian inferring, it can be factorized as:

$$p(L|\mathbf{X}, \mathbf{Z}) = \frac{p(\mathbf{Z}|L, \mathbf{X})p(L|\mathbf{X})}{p(\mathbf{Z}|\mathbf{X})} \propto p(\mathbf{Z}|L, \mathbf{X})p(L|\mathbf{X}). \tag{4}$$

In this paper, two main observations $\mathbf{Z}$ are considered: a spatial distribution $\mathbf{Z}_S$ based on the corner model and a color distribution $\mathbf{Z}_C$. As soon as an occlusion between tracked objects is detected, object models for each of these objects are initialized in the frame prior to the occlusion. The object models are updated during the whole occlusion period (see Section II-D4). We assume that the two observations are independent of each other. This lets us write:

$$p(L|\mathbf{X}, \mathbf{Z}) \propto p(\mathbf{Z}_S|L, \mathbf{X})p(\mathbf{Z}_C|L, \mathbf{X})p(L|\mathbf{X}). \tag{5}$$

The two observation distributions and the shape-prior distribution $p(L|\mathbf{X})$ will be described in the following subsections.

1) Color Observation: When computing the likelihood of the color observations $p(\mathbf{Z}_C|L, \mathbf{X})$, we consider each object as a whole and evaluate the observation at each pixel with the color distribution of an object that contains no position information:

$$p(\mathbf{Z}_C|L, \mathbf{X}) = p(\mathbf{Z}_C|L) = \mathbf{CD}_{\mathbf{Z}_C} \tag{6}$$

2) Spatial Observation: The corner points extracted to describe the object i are tracked from frame to frame using the Kanade Lucas Tomasi (KLT) algorithm [13]. This computes new corner positions $\mathbf{X}_{i,j}^t$ from the corner positions $\mathbf{X}_{i,j}^{t-1}$ in the previous frame. These tracked feature-points are used to describe the spatial distribution of the object. More specifically, we build a mixture-of-Gaussian model in which each tracked corner point corresponds to a mode of the Gaussian mixture with a constant covariance Σ and mean $\mu_{i,j} = \mathbf{X}_{i,j}^t$. Consequently, the likelihood of the spatial observation for the i-th object can be formulated as

$$p(\mathbf{Z}_S|L = l_i, \mathbf{X}) \propto \frac{1}{m}\sum_{j=1}^{m} N(\mathbf{X}|\Sigma, \mu_{i,j}). \tag{7}$$

Each mode is weighted equally. Figure 4(b) visualizes an example of this spatial distribution.

3) Shape-Prior Distribution: With the updated coordinates of the corner points, the new object position $\mathbf{P}_i^t$ can be estimated in a voting space. We use a two dimensional voting space corresponding to the pixel coordinates of the image. At the initialization, the values at all positions are set to zero. The voted position of each corner point is computed according to

$$\mathbf{Vote}_j^t = \mathbf{X}_{i,j}^t - \mathbf{S}_{i,j}^{t-1}, \quad j = 1 \dots m. \tag{8}$$

For every vote, the value of the voted position and its neighborhood (for the stability of the algorithm) is increased by one. The position in the voting space with the maximal value is chosen as the new center $\mathbf{P}_i^t$ of the object.

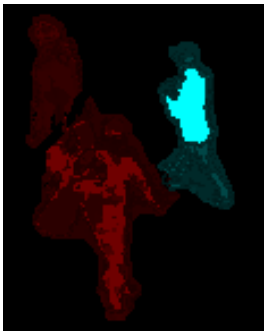
The estimated object motion $\Delta_i = \mathbf{P}_i^t - \mathbf{P}_i^{t-1}$ combined with the object mask from the last frame M_i^{t-1} defines the prior distribution $p(L|\mathbf{X})$

$$p(L = l_i|\mathbf{X}) = \begin{cases} 1/N(\mathbf{X}) & \mathbf{X} - \Delta_i \in M_i^{t-1}, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

where $N(\mathbf{X}) = |\{j|\mathbf{X} - \Delta_j \in M_j^{t-1}\}|$ is the number of objects that are predicted to cover pixel $\mathbf{X}$. In other words, pixels that are covered by N objects will receive a likelihood of $1/N$ for all these objects. Pixels that are not covered by any object receive likelihood zero.

(a) Original image.

(b) Weight map of objects based on the tracked corner points.



(c) Weight map based on color and spatial distributions.

(d) The final label image based on the fused weights according to Equation (5).

Figure 4. Example of the object model based on multiple cues.

With all the defined distributions above, Equation (5) is applied to assign to each pixel the object label with the maximum probability. In Figure 4(c) and 4(d), the result fusing the spatial and color distributions is shown. Thus, the merged observations are decomposed and the tracked objects can be updated with the assigned observations.

4) Model Update: To adapt the object models to changes of appearance, they are updated during the occlusion period. A new color distribution $\mathbf{CD}_{cur}$ is computed at the new estimated object position and the color model is updated with a given update ratio α as:

$$\mathbf{CD}' = (1 - \alpha)\mathbf{CD} + \alpha\mathbf{CD}_{cur}$$

The spatial features are re-computed with the new corner features and the new estimated object

III. Experimental Results

In this section, experimental results obtained with the described tracking method will be presented. The algorithm is tested on 25 sequences with approx. 35000 frames from PETS, ETISEO [10], CAVIAR [4] and other test sequences. These sequences have different difficulties. A tracking example with the proposed method is shown in Figure 5. To obtain an objective evaluation of the proposed method, a quantitative analysis is performed in the form of a comparison between the Algorithm Results (AR) and Ground-Truth (GT) with given measurements [3]. The ground-truth was obtained by manual labeling. Two aspects are most relevant for the tracking system: detection rate and uniqueness of tracking.

ObjectDetection FScore (DF) assesses the ability of the algorithm to detect objects frame-wise requiring that each object is detected separately, i.e., how well are the objects of interest detected and located? This means it simply reports the locations and number of objects in the scene without considering their identities. This *F-Score* is a harmonic mean of ObjectDetectionPrecision (DP) and ObjectDetectionSensitivity (DS). It is defined as

$$DF = \frac{2 \cdot DP \cdot DS}{DP + DS} \tag{10}$$

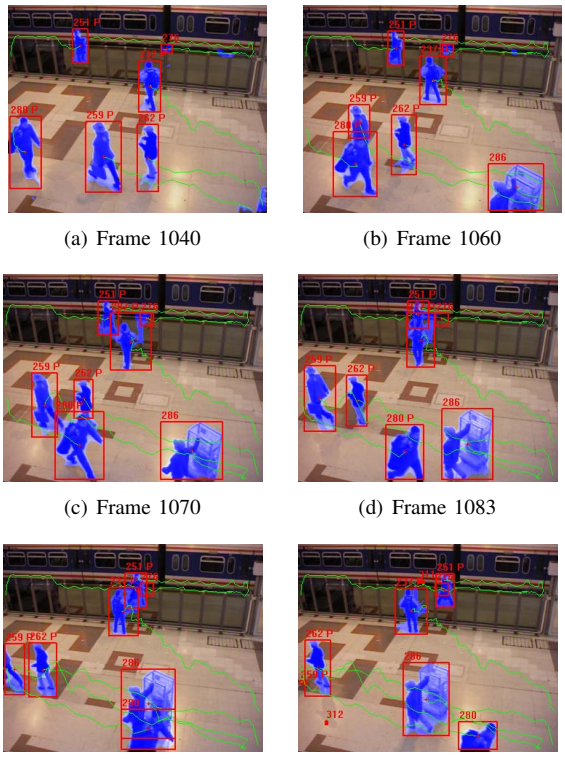
with

$$DP = \frac{ObjectDetTP}{ObjectDetTP + ObjectDetFP}, \tag{11}$$

$$DS = \frac{ObjectDetTP}{ObjectDetTP + ObjectDetFN}, \tag{12}$$

where $ObjectDetTP$ is the number of GT objects having a corresponding result object, $ObjectDetFP$ is the number of result objects *not* having a corresponding GT object and $ObjectDetFN$ is the number of GT objects *not* having a corresponding result object.

Purity FScore (PF) assesses the ability of the system to detect and track objects over time without losing them or changing their ID. It describes the persistence of tracking of



(a) Frame 1040 (b) Frame 1060

(c) Frame 1070 (d) Frame 1083

(e) Frame 1095 (f) Frame 1106

Figure 5. Tracking results with the test sequences PETS2006_S1-T1-C-3. The numbers above the bounding boxes indicate Track-IDs of objects. The blue areas are the segmented regions based on the background subtraction and the green lines are the tracking trajectories.

an object by a particular estimate over time and concerns with both spatial and temporal relations between ground truth and identified objects.

$$TrackerPurity(TP) = \frac{\#\text{correct frames } AR_{tr}(i)}{\#\text{frames } AR_{tr}(i)} \quad (13)$$

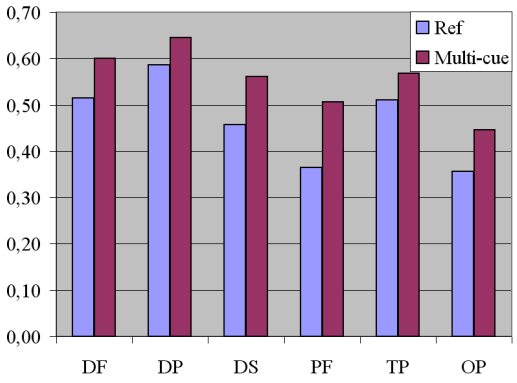
where $\#frames_{AR}(i)$ is the number of frames of the i-th *AR* track and $\#correct\ frames_{AR}(i)$ is the number of frames in which the i-th *AR* track corresponds to a *GT* track correctly.

$$ObjectPurity(OP) = \frac{\#\text{correct frames } GT_{tr}(i)}{\#\text{frames } GT_{tr}(i)} \quad (14)$$

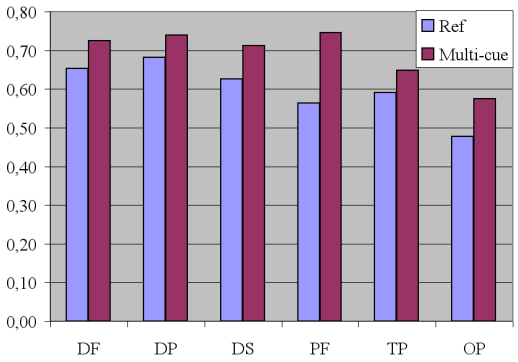
where $\#frames\ GT_{tr}(i)$ is the number of the frames of the i-th *GT* track and $\#correct\ frames\ GT_{tr}(i)$ is the number of frames in which the i-th *GT* track is correctly identified to an *AR* track.

$$PurityFscore(PF) = \frac{2 \cdot TrackerPurity \cdot ObjectPurity}{TrackerPurity + ObjectPurity} \quad (15)$$

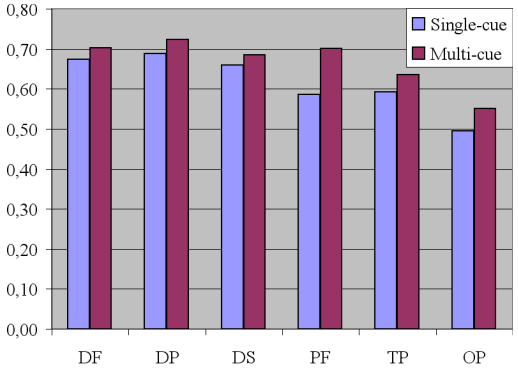
A tracking system that is based only on background-modeling is taken as the reference version. Instead of occlusion tracking, a post-processing that assigns the ID for



(a) Comparison between the reference tracking system and the proposed multi-cue based method on the indoor test sequences.



(b) Comparison between the reference tracking system and the proposed multi-cue based method on the outdoor test sequences.



(c) Comparison between the single cue (corner tracking) and multi-cue tracking system on all of the test sequences.

Figure 6. Comparison between the proposed multi-cue tracking system, a system based only on corner tracking, and the reference system. Six measures are used: object detection FScore (DF), object detection precision (DP), object detection sensitivity (DS), purity FScore (PF), tracker purity (TP) and object purity (OP).

occluded objects by comparing some simple object features, such as motion or object size, is used in the reference tracking system. It is compared to the proposed multi-cue based tracking system. The results are shown in Figure 6. The comparison is carried out for both indoor and outdoor sequences. The indoor subset consists of 8 sequences (7 CAVIAR sequences and a Bosch test sequence) and the outdoor subset consists of 17 sequences (7 ETISEO sequences, 6 CAVIAR sequences, 2 PETS2001 sequences, and 2 Bosch test sequences). We can see that a significant improvement is achieved both in the indoor and outdoor subsets. The indoor subset sequences are more challenging since more severe occlusions occur. A comparison between multi-cue based and single-cue, i.e. corner tracking, based tracking systems is also carried out. All measures are improved due to the integration of multiple cues. With the proposed method, the robustness of the tracking performance is increased, even when one of the introduced features is not distinctive enough. However, the proposed method does not work well in dense crowded scenes, where a reasonable background subtraction is no longer possible.

IV. Conclusions

In this paper, we have proposed a tracking system that integrates a corner-tracking with a color and shape model to overcome the *merge* problem in tracking occluded objects in cluttered scenes. The occlusion problem is handled well by the proposed method in moderately crowded scenes and the total performance of the tracking system is significantly improved. However, further work should be done to adapt the method for densely crowded scenes. Furthermore, additional features will be introduced to enhance the performance and an automatic selection and weighting of features will be studied. The current tracking system is still dependent on the segmentation based on background subtraction. A combination with a general object detector can make the system more flexible.

References

[1] M. Asadi, A. Dore, A. Beoldo, and C. Regazzoni. Tracking by using dynamic shape model learning in the presence of occlusion. In *AVSS*, 2007. 1, 2

[2] M. Asadi and C. S. Regazzoni. A probabilistic bayesian framework for model-based object tracking using undecimated wavelet packet descriptors. In *AVSS*, 2008. 1

[3] A. Baumann, M. Boltz, J. Ebling, M. Koenig, H. Loos, M. Merkel, W. Niem, J. Warzelhan, and J. Yu. A review and comparison of measures for automatic video surveillance systems. *JIVP*, 2008(2008), 2008. 4

[4] CAVIAR. http://homepages.inf.ed.ac.uk/rbf/caviar/. funded by the EC's Information Society Technology's programme project IST 2001 37540, 2004. 4

[5] D. Comaniciu, V. Ramesh, P. Meer, S. Member, and S. Member. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:564–577, 2003. 1

[6] A. Elgammal, R. Duraiswami, and L. Davis. Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking. *PAMI*, 25(11):1499–1504, November 2003. 1

[7] Y. Huang and I. Essa. Tracking multiple objects through occlusions. In *CVPR*, 2005. 1

[8] Z. Kim. Real time object tracking based on dynamic feature grouping with background subtraction. In *CVPR*, 2008. 1

[9] S. Mueller-Schneiders, T. Jaeger, H. S. Loos, and W. Niem. Performance evaluation of a real time video surveillance system. In *VS-PETS*, 2005. 2

[10] A. Nghiem, F. Bremond, M. Thonnat, and V. Valentin. Etiseo, performance evaluation for video surveillance systems. pages 476–481, 2007. 4

[11] C. S. Regazzoni, M. Asadi, and F. Monti. Feature classification for robust shape based collaborative tracking and model updating. *Journal on Image and Video Processing*, 2008, 2008. 1

[12] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer. Pedestrian detection and tracking for counting applications in crowded situations. In *AVSBS*, pages 70–70, 2006. 1

[13] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good features track better. *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 178–183, Jun 1998. 1, 3

[14] D. Xu, Y. Wang, and J. An. Applying a new spatial color histogram in mean-shift based tracking algorithm. In *Image and Vision Computing New Zealand (IVCNZ)*, 2005. 1

[15] L. Zhu, J. Zhou, and J. Song. Tracking multiple objects through occlusion with online sampling and position estimation. *Pattern Recognition*, 41(8):2447–2460, Aug. 2008. printed. 1