

# A Region-Based Object Tracking Scheme Using Adaboost-based Feature Selection

Fan-Tung Wei, Sheng-Ting Chou

Department of Computer Science & Information Engineering  
National Chung Cheng University  
Chiayi 621, Taiwan  
{wft94,cts95m}@cs.ccu.edu.tw

Chia-Wen Lin

Department of Electric Engineering  
National Tsing Hua University  
Hsinchu 30013, Taiwan  
cwlin@ee.nthu.edu.tw

**Abstract**—This paper presents an object extraction system for video surveillance applications that require pixel-wise extraction accuracy. The proposed mechanism is composed of two trackers. The first tracker extract video objects by using Adaboost on pixel-based global seed features, it can provide more detailed segmentation of target. The second tracker applies bidirectional labeling on regions as well as uses Adaboost on region-based local seed features to refine the object masks obtained from the first tracker. The system is featured with an interactive tool which allows users to deal with serious object occlusion situations. A confidence measure is proposed to minimize the effort of human interactions.

## I. INTRODUCTION

In many object tracking applications, objects are represented in some primitive geometric shapes, such like ellipses [1-3] and/or rectangles [4-7]. This kind of representation is simple to gather feature values from targets, and easy to be modified by translation, affine, or projective transformation. However, this kind of representation is not well suited for characterizing non-rigid targets. On the other hand, there are some more accurate representations of target, such as template [3,7], skeleton, and more complicated models [8]. Those representations usually have much detailed information of a target, but suffer from high complexity. Furthermore, since they are specific on particular targets, it is hard to extend them for general purposes.

Once the representation scheme is chosen, we tend to find the target in incoming frames to achieve tracking. An intuitive thought is to find the most similar one in the incoming frames. One can define some novel probabilistic likelihoods as kernel functions to characterize an incoming frame as a distribution, after that, we can find the optimal match using manifold mechanisms. For example, Lanz proposed a hybrid joint-separable model to cooperate with particle filtering [8].

On the other hand, we can use non-target regions to help find out the target [1,4,5]. The main difference from above is that we explicitly use samples as opposites but not just as conceptual dissimilarity. In such scenario, we treat tracking

problems as classification problems. After defining target and non-target samples, we collect them in incoming frames and tending to separate them from each other, by classification schemes or more complicated mechanisms. This kind tracking strategy usually works well due to richer features are available to be extracted, especially when training is available. In [3], Avidan used an optic-flow based support vector machine to train and classify vehicles. However, for non-rigid target applications, it is difficult to define such opposite samples; also there are lacks of time and samples for training when we talk about real-time and on-line needs.

To provide more detailed segmentation for further analysis, many tracking methods requires a background model [2,3]. These methods gain favors from useful background information but also be restricted on innate needs of backgrounds. Specifically, if the background is not still and/or time-varying so that it cannot be built correctly, those trackers will fail. Seed features mechanism [4], on the other hand, provides an opportunity to segment well without establishing any background model.

This work aims at offering a tool to extract video objects with pixel-wise accuracy, rather than characterizing objects with simple geometric shapes, so that the tool can be used in video editing and video surveillance applications that require accurate segmentations. Based on the Adaboost feature selection approach proposed in [4], we propose a novel dual-tracker algorithm to achieve pixel-wise object extraction.

The rest of this paper is organized as follows. Section 2 elaborates the proposed object extraction algorithm. The experimental results are shown in Section 3. Finally, conclusion is drawn in Section 4.

## II. PROPOSED METHOD

Fig. 1 shows the framework of the proposed algorithm. Initially, to reduce the computational complexity, the target object is located with a simple bounding rectangle as ROI (region of interest) at the first frame manually. The incoming frame along with the ROI information is fed into the dual-tracker module which consists of the pixel-wise tracker and

region-wise tracker. These two trackers are both built on top of the Adaboost feature selection algorithm [4], while one uses pixel-wise features and the other uses region-wise features. The region-wise tracker adopts a  $K$ -means clustering scheme to compensate for shortcoming of the pixel-wise tracker. The two object masks obtained from both trackers are post-processed using a morphological filter and combined to obtain the final object mask.

Since it is very difficult, if not impossible, to deal with any object occlusion situation while tracking/extracting an object using existing automatic tracking/extractation methods, our approach provides manual refinement tools to handle failure situations due to object occlusions. We propose a method of automatically identifying unreliable object extractions to trigger human interactions so as to minimize the effort of human interaction.

#### A. Pixel-wise Tracker

This tracker is mainly modified from the method described in [4]. Through the seed features selection and discriminability evaluation mechanism, the fittest features for test sequences can therefore be obtained.

We define our seed features as:

$$F = \{w_1R + w_2G + w_3B \mid w_i \in [-2, -1, 0, 1, 2], i = 1, 2, 3\} \quad (1)$$

where each corresponding weight  $w_i$  is chosen as an integer ranging from +2 to -2 for the three color components:  $R$  (red),  $G$  (green), and  $B$  (blue), respectively.

In order to separate foreground from background, we also use the tuned feature defined in (2) as an integrated expression and define thresholds to separate them from each other.

$$L(i) = \log \frac{\max\{p(i), \delta\}}{\max\{q(i), \delta\}} \quad (2)$$

where  $p(i)$  is the distribution of foreground and  $q(i)$  is that of background,  $i$  denotes the bin index, and  $\delta$  is a small value to prevent dividing by zero or taking the log of zero.

Every seed feature generates a corresponding tuned feature, which is then used for classifying foreground and background pixels. The hypothesis for pixel classification is defined as:

$$H(i) = \begin{cases} foreground & L(i) > \theta_{obj} \\ background & otherwise \end{cases} \quad (3)$$

where  $\theta_{obj}$  denotes the threshold to separate a foreground object from background. Our method uses three different thresholds to expand the solution space as follows:

$$\theta_{obj} = \{\max\{p(i)/2, \min\{-q(i)/2, 0\}\} \quad (4)$$

The value zero is a trivial threshold where the feature values of foreground pixels equal to those of background pixels. The other two thresholds represent different ratios of

predominance. Through this multi-threshold mechanism, we have a total of 147 features to construct the solution space.

Fig. 2 illustrates an example of thresholding the tune features.  $\theta_{obj} = \max\{p(i)/2, 0$  and  $\min\{-q(i)/2\}$  are represented in green, blue and yellow, respectively and the tuned feature is shown in dark-red line, respectively. The background and foreground are represented in orange and in cyan, respectively.

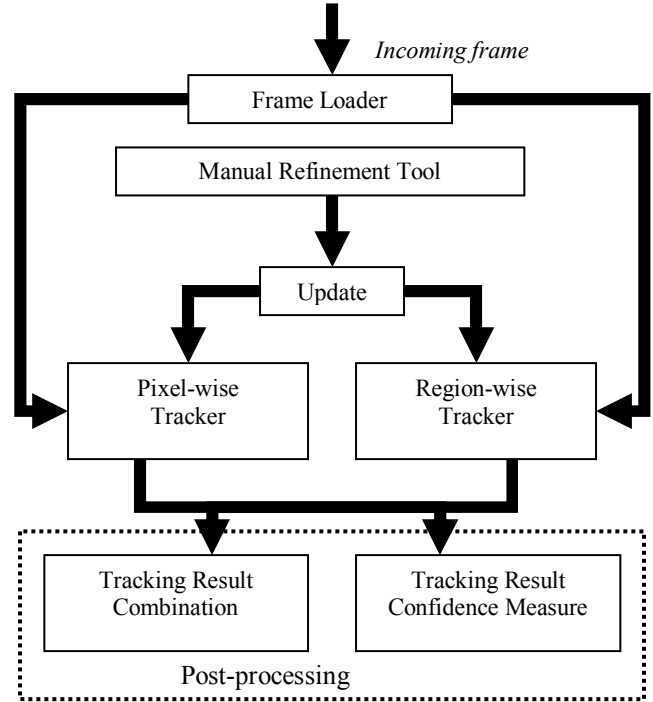


Fig. 1. Framework of the proposed method.

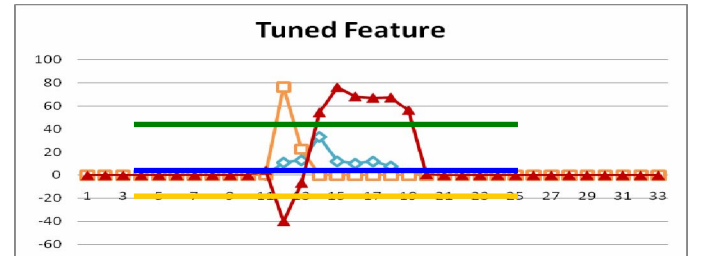


Fig. 2. Three thresholds on the tuned features.

We use the Adaboost algorithm to combine the tuned features, attempting to find much accurate segmentation result from them. One major advantage of this tracker is that seed features are capable of surviving during occlusion. Different from typical tracking methods that represent targets in simple geometric shapes, seed features has natural color thresholds instead of these artificial bounds, thus it has a potential to accurately extract targets from the frame, even if there exist serious occlusions as shown in Fig. 3. Because there usually does not exist a perfect threshold, we choose a threshold at 10%, meaning that if the confidence of a color/ region is not higher enough, the region is classified as background.

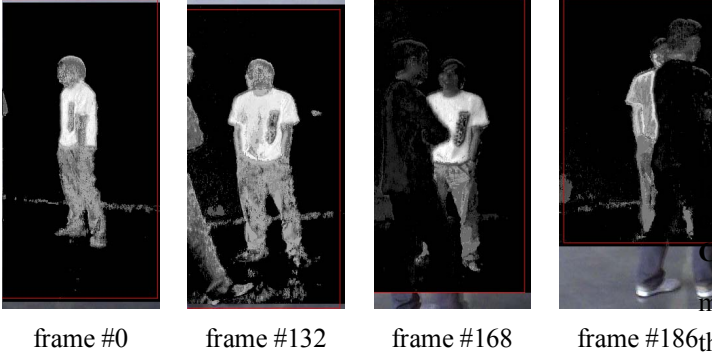


Fig. 3. An example of object classification using seed features with Adaboosting.

### B. Region-wise Tracker

The proposed pixel-wise tracker, however, may still result in false-positive and false-negative classifications. We propose a region-wise tracker, which consists of two  $K$ -means clustering operations followed by a post-processing procedure as elaborated below:

1) *Regionalization*: the tracker first performs  $K$ -means clustering to achieve regionalization. For each frame, every pixel  $\mathbf{p}$  is represented in a five-dimensional feature vector,

$$\mathbf{p} \equiv [x \ y \ R \ G \ B] \quad (5)$$

where  $x$  and  $y$  are its location coordinate of x-axis and y-axis, respectively.  $R$ ,  $G$ , and  $B$  represent its three color components. A pixel in the frame is classified into mean  $\mathbf{r}_i$  if  $\mathbf{r}_i$  is the closest color. The distance between the mean and the pixel is defined as follows:

$$D(\mathbf{p}, \mathbf{r}_i) = (1 - \alpha) D_{\text{location}}(\mathbf{p}, \mathbf{r}_i) + \alpha D_{\text{color}}(\mathbf{p}, \mathbf{r}_i), \quad 0 \leq \alpha \leq 1 \quad (6)$$

where  $D_{\text{location}}$  and  $D_{\text{color}}$  denote the spatial-location distance and the color distance, respectively, which are measured in the Euclidean distance as follows:

$$D_{\text{location}}(\mathbf{p}_1, \mathbf{p}_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (7)$$

$$D_{\text{color}}(\mathbf{p}_1, \mathbf{p}_2) = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (8)$$

To reduce classification complexity and regionalize pixels, we define the candidate region set for each pixel:

$$\mathfrak{S}_i \equiv \{\mathbf{r}_j | D_{\text{location}}(\mathbf{p}_i, \mathbf{r}_j) < d, \forall j\} \quad (9)$$

In (6),  $\alpha$  represents the coefficient to balance these two different kinds of distances. It plays an important role in regionalization. A large  $\alpha$  implies color consistency is emphasized while localization constrain is relaxed. We define a measure of color variety as:

$$\text{var}(\mathfrak{S}) \triangleq \frac{1}{3n_i} \sum_{\mathbf{r}_j \in \mathfrak{S}_i} \left[ (R_j - \bar{R})^2 + (G_j - \bar{G})^2 + (B_j - \bar{B})^2 \right] \quad (10)$$

where  $n_i$  represents the number of pixels in  $\mathfrak{S}$ . If  $\text{var}(\mathfrak{S})$  is large, we increase  $\alpha$  to stress on color consistency.

2) *Region tracking by bi-directional labeling*: this step tracks each region by finding the correspondance of the region between the current and previous frames using bi-directional labeling. The tracking can be divided into the following three cases:

**Case 1:** region  $\mathbf{r}_j^t$  in the current frame corresponds to one or more regions  $\{\mathbf{r}_i^{t-1}\}$  in the previous frame after performing the forward labeling in (11), and all these corresponding regions have the same label. If so, we consider  $\mathbf{r}_j^t$  has the same label as that of  $\{\mathbf{r}_i^{t-1}\}$ .

$$\text{label}(\mathbf{r}_j^t) = \left\{ \text{label}(\mathbf{r}_i^{t-1} | D(\mathbf{r}_j^t, \mathbf{r}_i^{t-1}) = \min_k D(\mathbf{r}_j^t, \mathbf{r}_k^{t-1})) \right\} \quad (11)$$

**Case 2:** for  $\mathbf{r}_j^t$ , if more than one forward corresponding region  $\{\mathbf{r}_i^{t-1}\}$  are found using forward labeling, but the labels of  $\{\mathbf{r}_i^{t-1}\}$  are not consistent, we apply the backward labeling in (12) to find the backward corresponding region and label  $\mathbf{r}_j^t$  accordingly.

$$\text{label}(\mathbf{r}_j^t) = \left\{ \text{label}(\mathbf{r}_i^{t-1} | D(\mathbf{r}_j^t, \mathbf{r}_i^{t-1}) = \min_k D(\mathbf{r}_j^t, \mathbf{r}_k^{t-1})) \right\} \quad (12)$$

**Case 3:** if no corespondiong region of region  $\mathbf{r}_j^t$  can be found in the previous frame by forward labeling,  $\mathbf{r}_j^t$  is labeled by (12).

The kernel distance function in this bi-directional lableing is similar to one in regionalization step:

$$D(\mathbf{r}_j^t, \mathbf{r}_i^{t-1}) = (1 - \beta) D_{\text{location}}(\mathbf{r}_j^t, \mathbf{r}_i^{t-1}) + \beta D_{\text{color}}(\mathbf{r}_j^t, \mathbf{r}_i^{t-1}) \quad (13)$$

3) *Region-based Adaboosting*: After labeling each region, we extract the mean color of each region, and use these color information to generate seed features. Again, Adaboosting is adopted in on-line learning of the region-based features.

### C. Post-Procesing and Manual-Refinement Tools

After obtaining the two labeled segmentation masks  $\phi_{\text{local}}$  and  $\phi_{\text{global}}$  using the pixel-wise and region-wise trackers, respectively, we perform morphological filtering to remove isolated label points. After the post-processing, the two filtered segmentation masks are combined to obtain the final results. The proposed algorithm can do a good job in extracting video objects most of the time. It, however, may lead to some incorrect segmented regions when serious object occlusion occurs which are in general very difficult to resolve using existing automatic segmentation techniques without any human interaction.

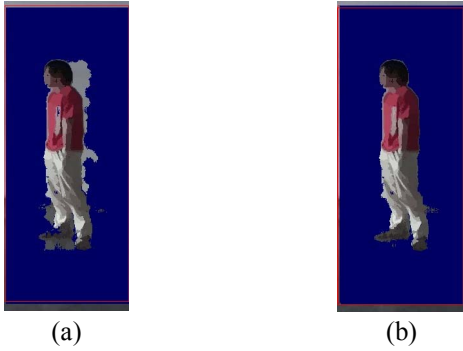


Fig. 4. (a) Original segmentation mask and (b) the result after manual refinement.

Therefore we provide two brush-like refinement tools to quickly refine the segmentation results. Users can simply drag this tool on the current frame to change several regions at the same time, and regions which are partially or fully passed through are adjusted accordingly. Fig. 4 illustrates an example of an object mask before and after using the manual-refinement tool. Note that, the result after performing manual refinement is very close to “ground truth” and thus can be fed back to update seed features coefficients in Adaboost.

It is not practical to perform manual refinement on each frame. To minimize the effort of manual interaction, we propose a confidence measure to automatically identify which frames are not segmented well and thus require manual refinement. We use the following XOR operation to calculate the mismatch between the pixel-wise and region-wise segmentation masks, which is considered the index of uncertainty.

$$u = \text{XOR}(\phi_{\text{local}}, \phi_{\text{global}}) = \sum_i \text{XOR}(\mathbf{p}_{\text{local}}^i, \mathbf{p}_{\text{global}}^i) \quad (14)$$

where  $\text{XOR}(\cdot)$  represents the binary exclusive OR function. A frame with a large  $u$  indicates that the final result is unreliable, thus manual refinement should be considered.

### III. EXPERIMENTAL RESULTS

In our experiments, the frame format of test sequences is 720×480 with interlacing. Figs. 5 and 6 show two of our test videos with and without camera motions, respectively. The first experiment is to track the right-hand-side person in Fig. 5. We can observe that the object extraction results (indicated in white color) are pretty good before object occlusion. During the occlusion period (see frame #150, #159), the mismatch mismatch value calculated by (14) becomes significantly large. Therefore, the manual-refinement tool is employed to confirm the real target for these frames, and the refined mask is fed into the Adaboost mechanism so as to update both trackers. The refined result has great improvement of confidence. Even with some serious occlusions, the proposed method may still be able to be tracked properly.

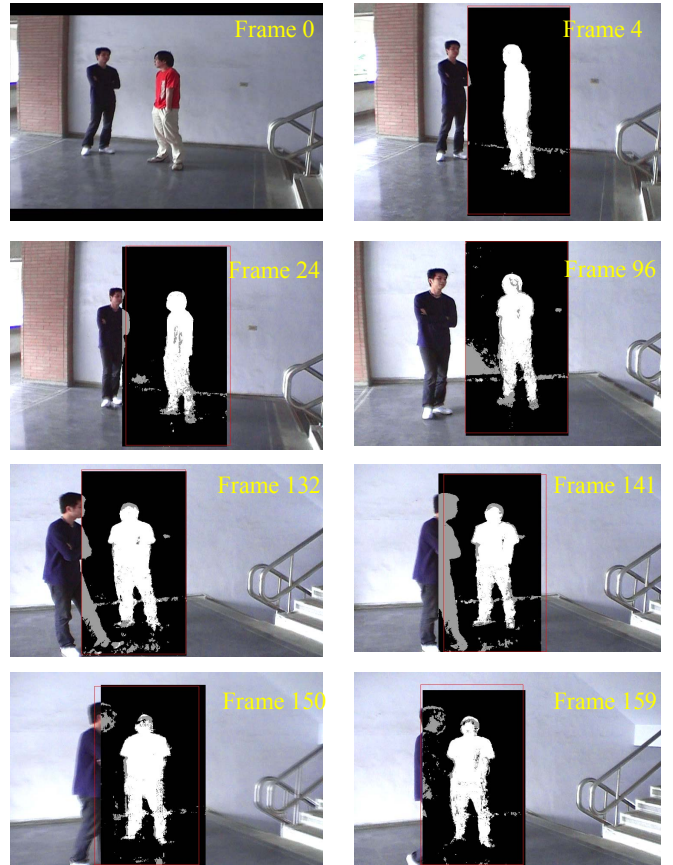


Fig. 5. Object extraction result of right-hand-side person in the *Two-Person* sequence. The white area indicates foreground pixels classified by both trackers; the gray area is the uncertain region; the black area is background.

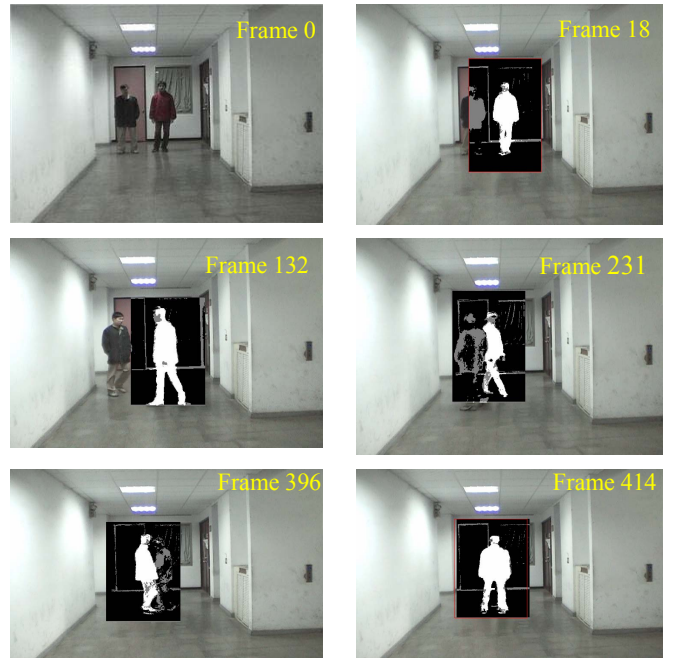




Fig. 6. Object extraction result of right-hand-side person in the Hall sequence using the proposed method.

We have tested another sequence as shown in Fig. 6. The sequence has no camera motion (i.e., the background is still), and the two people walk around each other which leads to significant changes in shapes and sizes of objects. We only track the right-hand-side person because he changes a lot and the left-hand-side person is almost the same behavior. In case of some occlusion regions that cannot be successfully segmented, the manual-refinement tool can be used to fix the results.

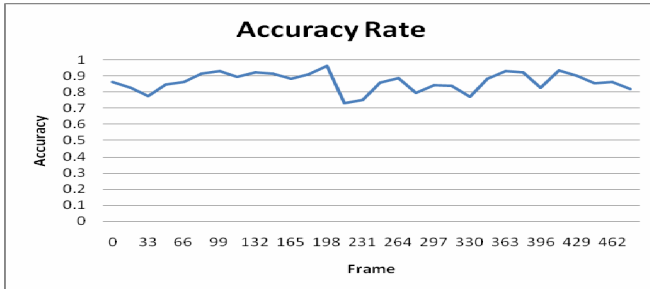


Fig. 7. Accuracy ratio with the proposed method for the Hall sequence.

#### IV. CONCLUSION

We proposed an object extraction scheme with pixel-wise accuracy. The proposed scheme mainly consists of two trackers. The pixel-wise tracker extracts an object using Adaboost-based global color feature selection. The region-based tracker first performs  $K$ -means clustering to regionalize each frame, and then achieves region tracking by using a bidirectional labeling scheme. Subsequently, the region-based tracker uses Adaboost with regional seed features to obtain region-wise segmentation result. Finally, the two object extraction results are fused to achieve accurate segmentation. Manual refinement tools are also provided to deal with serious occlusion situations. Experimental results show that that the proposed algorithm achieves robust object extraction even with some object occlusions.

#### ACKNOWLEDGEMENT

This work was supported in part by the Ministry of Economic Affairs (MOEA), Taiwan, R.O.C., under grant 96-EC-17-A-02-S1-032 and the National Science Council, Taiwan, R.O.C., under grant 95-2221-E-007-272-MY3.

#### REFERENCES

- [1] C. Hua, H. Wu, Q. Chen, and T. Wada, "A pixel-wise object tracking algorithm with target and background sample," in *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 739-742, Sept. 2006.
- [2] T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 26, no. 9, pp. 1208-1221, Sept. 2004.
- [3] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, vol. 2, pp. 406-413, July 2004.
- [4] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631-1643, Oct. 2005.
- [5] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064-1072, August 2004.
- [6] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261-271, Feb. 2007.
- [7] Y. Wu, T. Yu, and G. Hua, "Tracking appearances with occlusions," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, vol. 1, pp. 789-795, 2003.
- [8] O. Lanz, "Approximate Bayesian multibody tracking," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 28, no. 9, pp. 1436-1449, Sept. 2006.