Integrating Color and Shape-Texture Features for Adaptive Real-time Object Tracking

Junqiu WANG and Yasushi YAGI
The institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 560-0047
Telephone: +81-6-6879-8420 Fax: +81-6-6877-4375
Email: *jerywang@public3.bta.net.cn*, *yagi@am.sanken.osaka-u.ac.jp*

(Correspondence)

***Abstract*—We extend the standard mean-shift tracking algorithm to an adaptive tracker by selecting reliable features from color and shape-texture cues according to their descriptive ability. The target model is updated according to the similarity between the initial and current models, and this makes the tracker more robust. The proposed algorithm has been compared with other trackers using challenging image sequences, and it provides better performance.**

I. Introduction and Related Work

THE major difficulty in developing a visual tracker is the variation of the target appearance and its background. The appearance of a target tends to change especially during a long tracking process because of variations in illumination or viewpoint. The background in a long image sequence is dynamic even if it is captured by a stationary camera. The performance of a tracker can be improved by using an adaptive tracking scheme. The basic idea is to adaptively select features that make the object discriminative against its background. It is clear that multi-cue plays an important role in human visual perception. Therefore the integration of shape, texture, and color cues should result in better tracking performance against distractions. This work aims at developing an adaptive tracker by selecting discriminative features from multi-cue.

Recently several adaptive tracking algorithms [?], [?], [?], [?], [?], [?] have emerged. Stern and Efros [?] propose an algorithm in which the best feature is chosen from 5 color spaces. The foreground/background contrast is explicitly mentioned by Collins and Liu [?]. They propose switching the mean-shift tracking algorithm between various combinations of three color channels to select the color features that best distinguish the object histogram from the background histogram. In their work, only color features are used, which may not be discriminative enough under certain circumstances. There are also other adaptive tracking algorithms that use particle filters [?] or PCA based feature selection [?].

Adaptive tracking may fail due to model drifts [?]. During tracking, pixels in an image are classified as either target or background; however, this classification is not always perfect. Misclassified pixels cause the model to deviate from the real one. In [?], this problem is dealt with by computing the empirical object feature distribution in each frame by pooling pixel samples from the previously tracked image together with the labeled object pixels from the original training sample in the first frame which is assumed to be uncontaminated. They use the straightforward method of averaging the initial and current feature distributions. However, the model computed is not necessarily good enough for tracking because their approach involves an arbitrary update. We found that better performance could be achieved by using a novel updating strategy that takes into account the similarity between the initial and current appearance of the target. When the similarity is high, it is not necessary to compensate too much. Otherwise we should give different weights to the initial model and current appearance.

Multi-cue has been widely used in tracking and detection systems [?], [?], [?]. Haritaoglu and Flickner [?] track persons in image sequences captured by a stationary camera based on color and edge density with a mean-shift tracker. Isard and Blake [?] present a tracking method by combining a color model with a contour model. One of the drawbacks of this method though is that the construction of an explicit contour model is not easy. Birchfield [?] proposes a head tracking algorithm by modeling the head as an ellipse. The tracking is carried out by exhaustive searching in an image. Neither method adopts an adaptive scheme. The basic mean-shift algorithm [?], which is based on a color model, has achieved considerable success in object tracking because of its simplicity and robustness. However, color features cannot give good performance when an object and its background have similar colors. The color of an object depends on illumination, viewpoint and camera parameters that tend to change during a long tracking process. Fixed color features are therefore not always discriminative enough. Moreover, color histograms do not include other cues of the target that may be helpful for tracking.

We propose an adaptive tracking algorithm that represents the target using reliable features selected from color and shape-texture cues. Shape-texture cues can be represented by orientation histograms, which have been used effectively in gesture recognition [?]. The use of shape-texture cues has also achieved great success in conjunction with *Scale Invariant Feature Transformation (SIFT)* [?]. We have taken advantage of the shape-texture feature by adaptively combining it with color cue. These two cues are complementary under many circumstances [?].

II. Feature Extraction

A. Color Cue

Color distributions are represented by color histograms. We calculate color histograms in three color spaces: RGB, HSV and normalized rg. The R,G and B channels are quantized into 12 bins respectively. However, the RGB space is not enough for discrimination under all circumstances; therefore, the RGB space is converted into the HSV space. Hue and saturation are quantized into 12 bins. We found that intensity is less helpful in our tracking tasks and therefore the intensity factor is not used. The rg space has been shown to be reliable when the illumination changes. r and g are also used. There are thus 7 color features in the candidate feature set.

Hue and normalized rg are sensitive to noise since they are unstable near the achromatic axis [?]. We use a simple method to deal with this problem: the pixels near the achromatic axis are quantized into three gray bins according to their intensity values. Thus the random assignment of these unstable pixels is avoided.

A color histogram is calculated using a weighting scheme in which the Epanechnikov kernel is applied [**?**]. The background's histograms are built in the region around the target. The size of the region depends on the size of the target. The ratio of the two sizes (width and height) is 2:1 in this work. The weighting term is also used in building the background histograms.

B. Shape-Texture Cue

A shape-texture cue is described by an orientation histogram, which is computed based on image derivatives in *x* and *y* directions. We did not use the popular Sobel masks. Instead, the Scharr masks [**?**] (S_x and S_y) is used here because they give more accurate results than the Sobel kernel.

The gradients at the point (x, y) in the image *I* can be calculated by convolving the Scharr masks with the image. The gradients are defined as: $D_x(x, y) = S_x * I(x, y)$ and $D_y(x, y) = S_y * I(x, y)$. The strength of the gradient at the point (x, y) is computed as $D(x, y) = \sqrt{D_x(x, y)^2 + D_y(x, y)^2}$.

The orientation of the pixel is given by $\theta(x, y) = \arctan(D_y(x, y)/D_x(x, y))$. The orientations are also quantized into 12 bins. Each orientation is weighted and assigned to one of two adjacent bins according to its distance from the bin centers. This has been proven effective in [**?**].

III. Feature selection

A. Likelihood Ratio

The weighted histograms introduced in Section II do not reflect the descriptive ability of the features directly. A log-likelihood ratio image can be helpful in solving this problem [**?**], [**?**]. We get log-likelihood ratios based on the histograms of the foreground and background with respect to a given feature. The likelihood ratio produces a function that maps feature values associated with the target to positive values and those associated with the background to negative values. The frequency of the pixels that appear in a histogram bin is calculated as $\zeta_f^{(b_{in})} = p_f^{(b_{in})}/n_{fg}$ and $\zeta_b^{(b_{in})} = p_b^{(b_{in})}/n_{bg}$, where p_f and p_b are histogram n_{fg} is the pixel number of the target region and n_{bg} the pixel number of the background.

The log-likelihood ratio of a feature value is given by

$$L^{(b_{in})} = \max(-1, \min(1, \log\frac{\max(\zeta_f^{(b_{in})}, \delta_L)}{\max(\zeta_b^{(b_{in})}, \delta_L)})), \tag{1}$$

where δ_L is a very small number (δ_L is set to 0.001 in this work). The likelihood image for each feature is created by back-projecting the ratio into each pixel in the image.

Likelihood ratio images are the foundation for evaluating the discriminative ability of the features in the candidate feature set. Fig.**??** shows the likelihood ratio images of different features.

B. Feature Selection

Given m_d features for tracking, the purpose of the feature selection module is to find the best subset of features of size m_m, and $m_m < m_d$. Feature selection can help minimize the tracking error and maximize the descriptive ability of the feature set.

We find the features with the largest corresponding variances. Following the method in [**?**], based on the equality $\text{var}(x) = E[x^2] - (E[x])^2$, the variance of Equation.(**??**) is computed as

$$\text{var}(L; p) = E[(L^{b_{in}})^2] - (E[L^{b_{in}}])^2.$$

We evaluate the discriminative ability of each feature by calculating the variance ratio. In the candidate feature set, the color feature includes 7 different features: the color histograms of R, G, B, H, S, r, and g, while the shape-texture feature includes a gradient orientation histogram. These features are ranked according to the discriminative ability by comparing the variance ratio. The feature with the maximum variance ratio is taken as the most discriminative feature.

IV. Adaptive Mean-Shift Tracking

A. The Standard Mean-Shift Tracking Algorithm

The mean-shift algorithm is a robust non-parametric probability density estimation method for climbing density gradients to find the mode of the probability distributions of samples. It can estimate the density function directly from data without any assumptions about underlying distribution. This virtue avoids choosing a model and estimating its distribution parameters. The algorithm has achieved great success in object tracking [**?**], [**?**]. However, the basic mean-shift tracking algorithm assumes that the target representation is sufficiently discriminative against the background. This assumption is not always true especially when tracking is carried out in a dynamic background such as surveillance with a moving camera.

B. The Number of Features for Adaptive Tracking

After evaluation of the features in the candidate set, the features are ranked according to their discriminative ability against the background. Features with good discriminative ability can be combined to represent and localize the target. The combination of features needs to be carried out carefully. Intuitively, the more features we use, the better the tracking performance; however, this is not true in practice. According to information theory, a feature added into the system can bring negative effect as well as improvement in the performance [**?**]. This is due to the fact that the features used are not totally independent, and may correlated.

In our implementation, two features are used to represent the target. Based on the experimental results, this number has been found to be appropriate in most cases. We have tested systems using 1 or 3 features, but these gave inferior performances. The feature selection module runs every 8 to 12 frames. When the feature selection module selects features different from those in the initialization, only one feature is replaced each time. The second best feature of the previous selection will be discarded and replaced by the best one in the current selection. This strategy is very important in keeping the target from drifting.

C. Target Localization in Adaptive Tracking

The proposed tracking algorithm combines the best two features through back-projection [**?**] of the joint histogram, which implicitly contains certain spatial information that is important for the target representation. We calculate the joint histogram of the target with the best two features ($p_f^{(b_{in}^{(1)}, b_{in}^{(2)})}$) and a joint histogram of the searching region ($p_d^{(b_{in}^{(1)}, b_{in}^{(2)})}$).

We get a division histogram by dividing the joint histogram of the target by the joint histogram of the background,

$$p_d^{(b_{in}^{(1)}, b_{in}^{(2)})} = p_f^{(b_{in}^{(1)}, b_{in}^{(2)})}/p_b^{(b_{in}^{(1)}, b_{in}^{(2)})}. \tag{2}$$

The division histogram is normalized for the histogram back-projection. The pixel values in the image are associated with

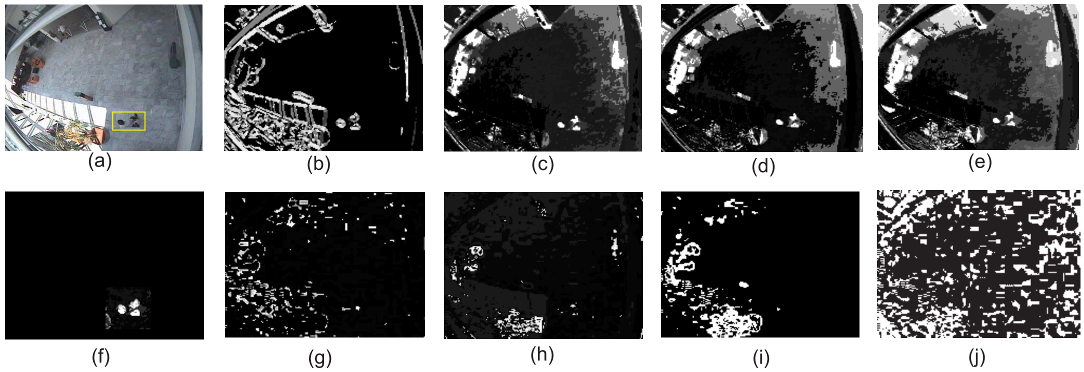

Fig. 1. (a) Original image. Likelihood ratio images of (b) the orientation histogram; (c) R; (d) G; (e) B; (f) integration of the best two features: orientation histogram and R; (g) H; (h) S; (i) r; and (j) g. Note that the likelihood image in (f) shows only the neighborhood of the target.

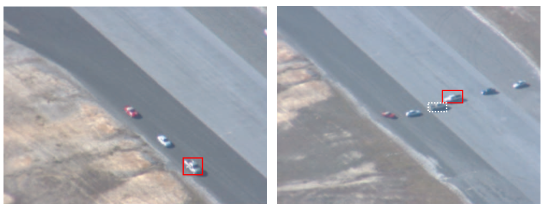

Fig. 2. Tracking the EgTest02 using the variance ratio algorithm [**?**]. The image on the left is the first frame and the image on the right is the frame where the variance ratio algorithm fails. The white bounding box in the right image is the ground truth.

the value of the corresponding histogram bin by histogram back-projection. The back-projection of the target histogram with any consecutive frame generates a probability image $p = \{p^i\}_{i=1 \ldots n_h}$ where the value of each pixel characterizes the probability that the input pixel belongs to the histograms. The two images of the best two features have been computed for the back-projection. Note that the H, S, r, and g images are calculated by transferring the original image to the HSV and the rg spaces; the orientation image has been calculated using the approach introduced in section II(B).

The initial position of the target is given by $\mathbf{y}_0$ which is determined in the last frame. Since we are using an Epanechnikov profile [**?**] the derivative of the profile, $g(x)$, is constant. The target's shift vector from $\mathbf{y}_0$ in the current frame is computed as

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i p^i(\mathbf{y}_0)}{\sum_{i=1}^{n_h} p^i(\mathbf{y}_0)}. \tag{3}$$

We compute Bhattacharyya coefficient [**?**] and the tracker assigns a new position to the target by using

$$\hat{\mathbf{y}}_1 = \frac{1}{2}(\hat{\mathbf{y}}_0 + \hat{\mathbf{y}}_1). \tag{4}$$

If $\|\hat{\mathbf{y}}_0 - \hat{\mathbf{y}}_1\| < \varepsilon$, the computation stops and $\mathbf{y}_1$ is taken as the position of the target in the current frame. Otherwise let $\hat{\mathbf{y}}_0 = \hat{\mathbf{y}}_1$. Then we compute the shift vector by using Equation. (**??**) and do position assignment by using Equation. (**??**) iteratively. If the computation does not converge after 15 loops, we stop the iteration and use the current location as the final result. In most cases, however, the algorithm converges in 3 to 6 loops.

D. Updating the Target Model

It is necessary to update the target model because the appearance of a target tends to change during a tracking process. Unfortunately, updating the target model adaptively may lead to tracking drift because of the imperfect classification of the target and background. Collins and Liu [**?**] proposed that forming a pooled estimate allows the object appearance model to adapt to current conditions while keeping the overall distribution anchored to the original training appearance of the object. They assume that the initial color histogram remains representative of the object appearance throughout the entire tracking sequence.

To update the target model, we propose an alternative approach considering the initial model, previous model and current candidate. During the initialization stage, the target is labeled and the initial target model (H_i) is computed. The initial target model is used for tracking in the next frame and is also kept to be used in subsequent updates. The weight of the initial model in the update is determined by the similarity between the initial model and the current target.

Before the i^{th} update, the tracker searches for the target in the current frame using the previously computed target model H_m^{i-1}. The target is localized by using the method given in the previous subsection. Then we compute the new target model H_m^i by using

$$H_m^i = (1 - s_{ic})H_i + s_{ic}H_c^i, \tag{5}$$

where H_i is the initial model; H_c^i the combined histogram of the current target appearance and the previous target model; and s_{ic} the similarity between the initial model and current target appearance. s_{ic} is measured by a simple correlation based template matching [**?**] performed on the current frame using the initial target appearance as the template. The similarity measure can be viewed as the alignment between the initial model and the current appearance of the target. Template matching is adopted here because it is based raw pixel comparison and can reflect the small differences. Since we do not use a large search window that is necessary in template matching-based tracking, the matching process is efficient and adds little computational cost to our algorithm.

H_c^i is computed by combining the current target appearance and the previous target model:

$$H_c^i = (1 - s_{mc})H_m^{i-1} + s_{mc}H_a, \tag{6}$$

where H_m^{i-1} is the previous target model; H_a is the histogram of current target appearance; and s_{mc} is the similarity between the previous target model and the current target appearance that is measured by Bhattacharyya coefficient [**?**].

The proposed updating method considers temporal coherence by weighting the initial target model, previous target model and current candidate. The variance ratio algorithm [**?**] relies on the initial model without considering the similarities. In Fig. **??**, we show the advantage of our updating method over the simple average method

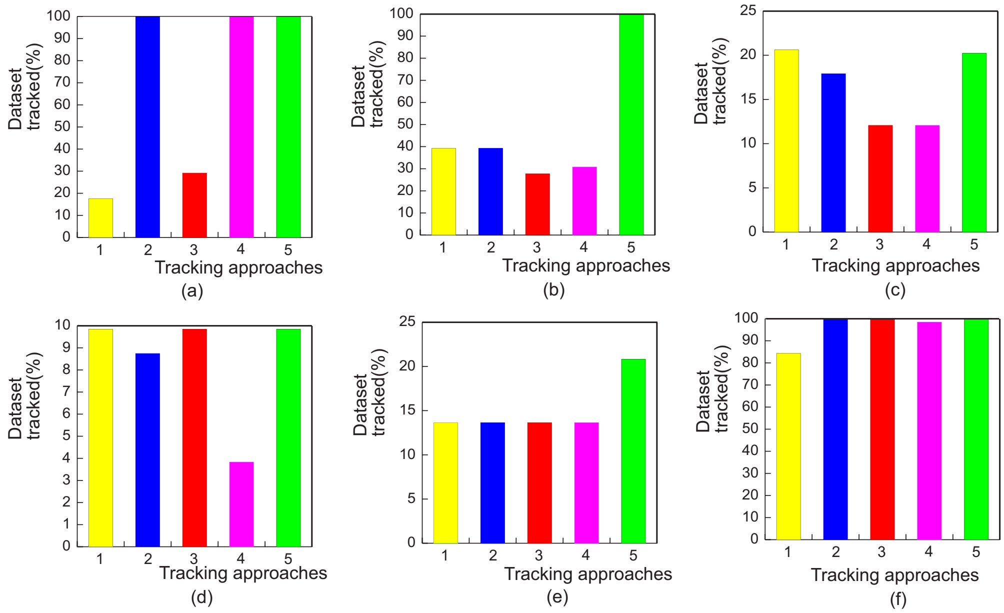

Fig. 3. Comparing the percentages of dataset tracked using different tracking approaches. Tracker 1 is the basic mean-shift algorithm; 2 the foreground/background ratio algorithm; 3 the variance ratio algorithm; 4 the peak difference algorithm; and 5, the proposed algorithm. Tests are performed on (a) EgTest01; (b) EgTest02; (c) EgTest03; (d) EgTest04; (e) EgTest05; and (f) redteam.

in [?]. The initial appearance of the target is similar to another car nearby in the right image. The variance ratio algorithm is attracted by the car because the initial model is added into the target model directly. The proposed algorithm can track the target thorough out the sequence thanks to the novel updating method. The details of performance evaluation will be described in the next section.

V. Experimental Results

To check the effectiveness of the proposed approach, we have implemented and tested it on a wide variety of challenging image sequences in different environments and applications. The tracking results are compared with those produced by the basic mean-shift and other algorithms. The current implementation runs 14 frames/sec on an Intel Centrino 1.4GHz laptop with 256MB RAM when applied to images of size 640×480. It achieves 33 frames/sec on an Intel Pentium D 3.0GHz PC with 1GB RAM. The execution times include the time taken for the main tracking algorithm, as well as the time taken for reading the image files from a USB disk and displaying the color images with the object bounding box overlaid.

A. Quantitative Performance Evaluation

To quantitatively evaluate the effectivity and accuracy of the proposed algorithm, we tested it on a public dataset with ground truth [?]. The tracking results are compared with the basic mean-shift [?], fore/background ratio [?], variance ratio and peak difference [?] trackers. The initialization of the tracking is given by mask images. We use the same initialization for all the trackers. The dataset includes 6 sequences: EgTest01 (1820 frames), EgTest02 (1300 frames), EgTest03 (2570 frames), EgTest04 (1832 frames), EgTest05 (1763 frames) and Redteam (1917 frames). There are various factors that make the tracking challenging: different viewpoints (these sequences are captured by moving cameras); illumination changes; reflectance variations of the targets; similar objects nearby, and partial occlusions.

Two types of metrics are used in the evaluation namely, the tracking success rate and the appearance accuracy of the computed foreground masks. The most important criterion is the percentage of dataset tracked, which is the number of tracked frames divided by the total number of frames. The track is considered to be lost if the bounding box does not overlap the ground truth. The comparison results are shown in Fig. **??**. The proposed tracker gives the best results in five of the test sequences and is second best in the remaining test, EgTest03, in which the best tracker has only a 0.39% advantage. Although an adaptive strategy is applied in the variance ratio algorithm [?], it does not give good performance (ranking 4^{th}) in three sequences: EgTest01, EgTest02 and EgTest03. Based on the variance ratio algorithm, the peak difference algorithm [?] has been developed in which distraction analysis is applied. It results in better performance in the first three sequences. However, its rankings in four sequences (EgTest02, EgTest03, EgTest04, EgTest05) are much lower than those of the proposed algorithm. These comparisons demonstrate that the proposed tracking algorithm has better performance than the other trackers.

The accuracy of tracking is evaluated by four criteria: average overlap between bounding boxes (Avg overlap BB), which is the percentage of the overlap between the tracking bounding box and the bounding box identified by ground truth files; average overlap between bitmaps within the overlapping bounding box area (Avg overlap BM), which is computed in the area of the intersection between the user bounding box and the ground truth bounding box; the average distance transform focused on the ground-truth object (Avg DT (US to GT)), which is the chamfer distance [?] between the bitmap of the target and the ground truth; the average distance transform is focused on the tracker identified object (Avg DT (GT to US)), in which the tracker-supplied bitmap is used to compute the distance transform against which the ground truth bitmap is scored. The comparison results are shown in Table **??**. The proposed tracking algorithm is not the best in some of the sequences; however, there is not much difference between the

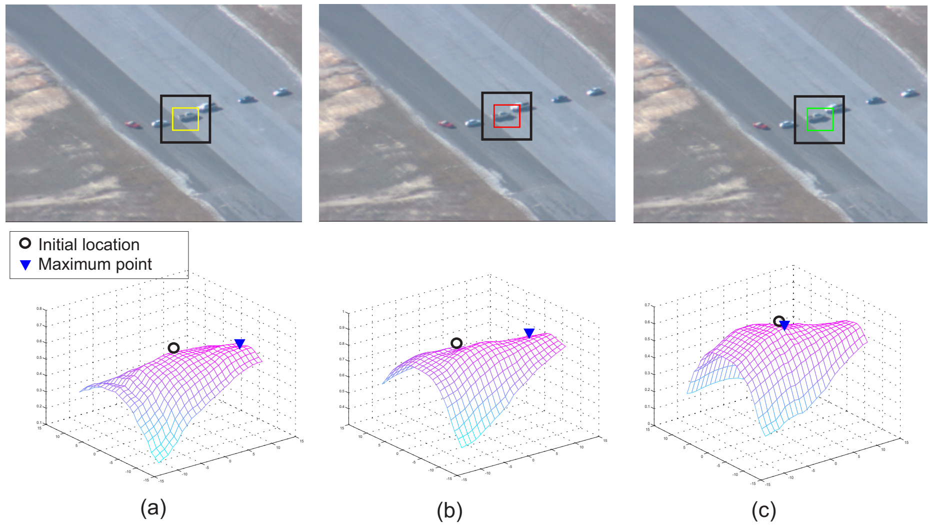

Fig. 4. The Bhattacharyya coefficients corresponding to the black rectangles in the first row are computed in frame 460 of EgTest02. The similarity surfaces of three algorithms are shown: (a) the basic mean-shift algorithm; (b) the variance ratio algorithm; and (c) the proposed algorithm.

TABLE I
QUANTITATIVE PERFORMANCE EVALUATION OF DIFFERENT TRACKERS.

(a) EgTest01.

Algorithms	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	TheProposed
Avg overlap BB	$65.50_{(2)}$	$62.87_{(3)}$	$76.87_{(1)}$	$61.76_{(4)}$	$61.62_{(5)}$
Avg overlap BM	$66.26_{(2)}$	$49.15_{(5)}$	$61.30_{(3)}$	$57.76_{(4)}$	$68.38_{(1)}$
Avg DT (US to GT)	$2.90_{(5)}$	$0.62_{(2)}$	$0.62_{(2)}$	$0.41_{(1)}$	$0.68_{(4)}$
Avg DT (GT to US)	$4.53_{(5)}$	$3.15_{(4)}$	$1.91_{(1)}$	$2.70_{(2)}$	$2.86_{(3)}$

(b) EgTest02.

Algorithms	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	TheProposed
Avg overlap BB	$91.09_{(2)}$	$89.13_{(4)}$	$85.19_{(5)}$	$90.54_{(3)}$	$93.32_{(1)}$
Avg overlap BM	$74.69_{(1)}$	$66.98_{(4)}$	$73.32_{(2)}$	$65.91_{(5)}$	$72.70_{(3)}$
Avg DT (US to GT)	$3.17_{(5)}$	$1.83_{(3)}$	$2.70_{(4)}$	$0.65_{(1)}$	$1.49_{(2)}$
Avg DT (GT to US)	$0.73_{(3)}$	$1.44_{(5)}$	$0.72_{(2)}$	$1.35_{(4)}$	$0.49_{(1)}$

(c) EgTest03.

Algorithms	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	TheProposed
Avg overlap BB	$86.96_{(5)}$	$87.01_{(4)}$	$93.74_{(1)}$	$92.27_{(2)}$	$88.66_{(3)}$
Avg overlap BM	$66.65_{(4)}$	$54.04_{(5)}$	$70.79_{(1)}$	$67.20_{(3)}$	$69.37_{(2)}$
Avg DT (US to GT)	$3.34_{(5)}$	$1.38_{(2)}$	$1.75_{(3)}$	$2.02_{(4)}$	$0.90_{(1)}$
Avg DT (GT to US)	$0.98_{(3)}$	$2.51_{(5)}$	$0.39_{(1)}$	$0.54_{(2)}$	$1.05_{(4)}$

(d) EgTest04.

Algorithms	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	TheProposed
Avg overlap BB	$66.78_{(3)}$	$67.92_{(2)}$	$66.03_{(4)}$	$63.60_{(5)}$	$69.52_{(1)}$
Avg overlap BM	$59.70_{(3)}$	$52.75_{(4)}$	$66.74_{(1)}$	$66.42_{(2)}$	$56.34_{(5)}$
Avg DT (US to GT)	$1.32_{(3)}$	$0.10_{(1)}$	$1.34_{(4)}$	$2.46_{(5)}$	$0.13_{(2)}$
Avg DT (GT to US)	$1.57_{(3)}$	$1.61_{(4)}$	$1.38_{(2)}$	$1.72_{(5)}$	$1.35_{(1)}$

(e) EgTest05.

Algorithms	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	TheProposed
Avg overlap BB	$94.58_{(1)}$	$88.75_{(2)}$	$86.46_{(4)}$	$86.98_{(3)}$	$72.65_{(5)}$
Avg overlap BM	$84.02_{(2)}$	$71.12_{(3)}$	$85.12_{(1)}$	$69.90_{(4)}$	$64.45_{(5)}$
Avg DT (US to GT)	$0.49_{(3)}$	$0.14_{(2)}$	$1.11_{(5)}$	$0.09_{(1)}$	$0.70_{(4)}$
Avg DT (GT to US)	$0.42_{(1)}$	$0.95_{(3)}$	$0.84_{(2)}$	$1.39_{(4)}$	$2.93_{(5)}$

(f) Redteam.

Algorithms	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	TheProposed
Avg overlap BB	$68.37_{(5)}$	$73.22_{(2)}$	$73.24_{(1)}$	$72.37_{(4)}$	$72.61_{(3)}$
Avg overlap BM	$75.05_{(3)}$	$51.13_{(5)}$	$75.30_{(2)}$	$78.54_{(1)}$	$55.58_{(4)}$
Avg DT (US to GT)	$1.74_{(5)}$	$0.38_{(1)}$	$0.71_{(4)}$	$0.63_{(3)}$	$0.52_{(2)}$
Avg DT (GT to US)	$2.36_{(4)}$	$2.40_{(5)}$	$1.68_{(1)}$	$1.68_{(1)}$	$2.11_{(3)}$

(a)

(b)

Fig. 5. Two failure examples of the proposed tracker. Example (a) shows the tracking result of EgTest03 and (b) that of EgTest05. The image on the left in each row is the first frame. The image on the right is the frame where the tracker fails. The dashed bounding boxes are the ground truth.

results of the best tracker and those of the proposed algorithm. The main reason for the unsatisfying tracking accuracy is that the two-dimensional histograms in the proposed tracker are relatively sparse and the back-projection is not always accurate enough. The accuracy of tracking of the proposed algorithm can be improved by using the *sigmoid M-estimator*.

To demonstrate the advantage of our approach further, Fig. **??** shows three similarity surfaces obtained by computing the Bhattacharyya coefficient [**?**]. The similarity surfaces of the mean-shift and variance ratio algorithms [**?**] are badly affected by the car near the target, which leads to tracking failures. In contrast to these traditional algorithms, the proposed approach gives a reasonable similarity surface and successfully tracks the target.

Failure Analysis: Although the proposed algorithm has better performance than other trackers it fails in some sequences. The main reasons leading to the failures include distraction by a very similar object nearby and long duration of heavy occlusions. In Fig. **??**(a), the tracker is attracted by the green car near the target; in Fig. **??**(b), the target is occluded by the trees and the tracker can not find it.

VI. Conclusion

We present an adaptive tracking algorithm by integrating shape-texture and color features in an adaptive way. The target is represented by a joint histogram of the best two features that implicitly includes certain spatial information. The proposed approach demonstrates good performance on challenging sequences.

References

[1] M. Atallah, "Faster image template matching in the sum of the absolute value of differences measure," *IEEE Trans. on Image Processing*, vol. 10, no. 4, April 2001.

[2] S. Avidan, "Ensemble Tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp.II: 20-25, 2005.

[3] H. G. Barrow. Parametric correspondence and chamfer matching: two new techniques for image matching. In *Int'l Joint Conf. on Artificial Intelligence*, pages 659-663, 1977.

[4] S. Birchfield, "Elliptical Head Tracking using Intensity Gradients and Color Histograms", in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 232-237, 1998.

[5] G.R. Bradski, "Computer Vision Face Tracking as a Component of a Perceptural User Interface," in *Proc. of the IEEE Workshop Applications of Computer Vision*, pp. 214-219, 1998.

[6] H.T. Chen, T.L. Liu and C.S. Fuh, "Probabilistic tracking with adaptive feature selection," In *Proc. of Int'l Conf. on Pattern Recognition*, pp. II: 736-739, 2004.

[7] R. T. Collins and Y. Liu, "On-line Selection of Discriminative Tracking Features," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp. 1631-1643, 2005.

[8] R. T. Collins, X. Zhou, and S. K. Teh, "An Open Source Tracking Testbed and Evaluation Web Site", in *IEEE Int'l Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005)*, January, 2005.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based Object Tracking," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 25, no. 5, pp. 564-577, 2003.

[10] T.M Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley and Sons Press, 1991.

[11] J. Wang and Y. Yagi. "Integrating shape and color features for adaptive real-time object tracking", *2006 IEEE Int'l Conf. on Robotics and Biomimetics*, 2006.

[12] T. Gevers and A.W.M Smeulders, "Color based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453-464, 1999.

[13] B. Han, L. Davis, "Object tracking by adaptive feature extraction," in *Proc. of the IEEE Conf. on Image Processing*, vol. 3, pp. 1501-1504, 2004.

[14] I. Haritaoglu and M. Flickner, "Detection and tracking of shopping groups in stores," in *Proc. of the IEEE Conf. on computer vision and pattern recognition*, vol. I, pp. 431-438, 2001.

[15] M. Isard and A. Blake, "Condensation - conditional density propagation for tracking," *Int'l Journal of Computer Vision*, vol. 29, no. 1, pp. 2-28, 1998.

[16] B. Jhne, H. Scharr, and S. Krkel. "Principles of filter design" In Jhne, B., Hauecker, H., and Geiler, P., editors, *Handbook of Computer Vision and Applications*, volume 2, pages 125-151. Academic Press, 1999.

[17] A. D. Jepson, D. J. Fleet, and T. EI-Maraghi. "Robust online appearance models for visual tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296-1311, 2003.

[18] A.P. Leung and S. Gong, "Online Feature Selection Using Mutual Information for Real-Time Multi-View Object Tracking," in *ICCV Workshop on AMFG*, 2005.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[20] H.T. Nguyen and A. Smeulders, "Robust tracking using foreground-background texture discrimination," *Int'l Journal of Computer Vision*, 69(3), pp. 277-293, 2006.

[21] H. Stern and B. Efros, "Adaptive Color Space Switching for Face Tracking in Multi-color Lighting Environment," in *Proc. IEEE Int'l Conf. on Automatice Face and Gesture Recognition*. Washington DC, USA, 2002, pp. 249-254.

[22] M. Swain and D. Ballard, "Color Indexing," *Int'l Journal of Computer Vision*, vol. 7, pp. 11-32, 1991.