Discriminative Mean Shift Tracking with Auxiliary Particles

Junqiu Wang and Yasushi Yagi

The Institute of Scientific and Industrial Research, OSAKA University
8-1 Mihogaoka, Ibaraki, Osaka, Japan
jerywangjq@gmail.com

Abstract. We present a new approach towards efficient and robust tracking by incorporating the efficiency of the mean shift algorithm with the robustness of the particle filtering. The mean shift tracking algorithm is robust and effective when the representation of a target is sufficiently discriminative, the target does not jump beyond the bandwidth, and no serious distractions exist. In case of sudden motion, the particle filtering outperforms the mean shift algorithm at the expense of using a large particle set. In our approach, the mean shift algorithm is used as long as it provides reasonable performance. Auxiliary particles are introduced to conquer the distraction and sudden motion problems when such threats are detected. Moreover, discriminative features are selected according to the separation of the foreground and background distributions. We demonstrate the performance of our approach by comparing it with other trackers on challenging image sequences.

1 Introduction

Tracking objects through image sequences is one of the fundamental problems in computer vision. Among the algorithms developed in the pursuit of robust and efficient tracking, two major successful approaches are the mean shift algorithm [1][5], which focuses on *Target Representation and Localization*, and particle filtering [7][9], which is developed based on *Filtering and Data Association*. Both of them have their respective advantages and drawbacks. This paper aims at developing a robust and efficient tracker that incorporates the efficiency of the mean shift algorithm with the multi-hypothesis characteristics of the particle filtering.

The mean shift algorithm is a robust non-parametric probability density estimation method. Comaniciu et al. [5] define a spatially-smooth similarity function and reduce the state estimation problem to a search of the basin of attraction of this function. Since the similarity function is smooth, a gradient optimization method leading to fast localization is applied. Despite its efficiency and robustness, the mean shift algorithm is not good at coping with quick motions. The distractions in the neighborhood of the target are threats to successful tracking. In addition, the basic mean-shift algorithm assumes that the target representation is sufficiently discriminative against the background. This assumption is

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 576–585, 2007.

not always true especially when tracking is carried out in a dynamic background such as surveillance with a moving camera. We introduce particles to deal with the first two problems because they are able to provide multiple hypothesis. Adaptive tracking is one possible solution to alleviate the third problem [3]. We update the target model according to the separation of the foreground and background distributions.

Particle filtering stands out in filtering-based techniques due to its ability to represent multi-modal probability distributions using a weighted sample set $S = \{(s^{(n)}, \pi^{(n)})|n = 1, \ldots, N\}$ that keeps multiple hypothesis of the states of targets [7] [9]. When the tracking is performed in a cluttered environment where multiple objects similar to the target can present, particle filters are able to find the target by validation and association of the measurements. However, since the number of particles can be large, a potential drawback of particle filtering is the high computational cost. Moreover, the particle set can degenerate and diffuse in a long sequence. Only few particles with high weights are useful after the tracking in certain frames. Accurate models of shape and motion learned from examples have been used to deal with these problems [9]. One of the drawbacks of this method though is that the construction of explicit models sometimes is hardly achievable because of viewpoint changes.

Blake et al. [10] proposed the ICONDENSATION algorithm in which high and low-level information are combined using importance sampling. However, it is complicated to model the dynamic characteristics accurately in an uncontrolled environment. Sullivan and Rittscher [14] noticed the advantages of the mean shift and particle filter algorithms. They proposed a particle filter-based tracking guided by deterministic search based on a SSD type cost function. The size of particle set is adjusted according to the difficulty of the problem at hand, which is indicated by motion. Deterministic search using mean-shift has also been applied in a hand tracking algorithm by embedding the mean-shift optimization into particle filtering to move particles to local peaks in the likelihood, which improves the sampling efficiency [13]. Although the mean-shift and particle filters have been combined in various ways in previous works, none of them deal with occlusions and distractions explicitly. Cai et al. [2] embed the mean-shift algorithm into the particle filter framework to stabilize the trajectories of the targets. It is necessary to learn classifiers for the targets in their work, which is not always possible in tracking applications.

The mean shift tracking algorithm outperforms the particle filter when the representation of a target is discriminative enough, the target does not jump beyond the bandwidth, and no serious distractions exist. Although it seems that these conditions are too strict, we observed that they can be met in a large percentage of real image sequences captured for surveillance or other applications. In this work, the mean-shift algorithm is adopted as the main tracker as long as these conditions are met. In other words, only one particle driven by the mean-shift searching is used to estimate the state of the target. Auxiliary particles are introduced when sudden motion or distractions are detected. We compute log likelihood ratios of class conditional sample densities of the target

and its background. These ratios are applied in feature selection and distraction detection. The target model is updated according to feature selection results. Sudden motions are estimated using the efficient motion filters [16].

The proposed method offers several advantages. It achieves high efficiency when the target moves smoothly. When sudden motions or distractions are detected, auxiliary particles are initialized to support the mean shift tracker. The help from particle filtering partially solves the problems resulted from sudden motions or distractions.

The remainder of the paper is organized as follows. Section 2 gives a brief introduction of the target model. Section 3 describes the feature selection and model updating methods. Section 4 introduces motion estimation and distraction detection. Section 5 discusses the use of auxiliary particles. The performance of the proposed method is evaluated in Section 6 and conclusions are given in Section 7.

2 Target Modeling

The target model should be as discriminative as possible to distinguish between complex target and background. We use an adaptive target model represented by the best features selected from shape-texture and color cues [17].

Color histograms are computed in three color spaces: RGB, HSV and normalized rg. There are 7 color features (R, G, B, H, S, r, g) in the candidate feature set. These color channels are quantized into 12 bins respectively. A color histogram is calculated using a weighting scheme in which the Epanechnikov kernel is applied [5].

A shape-texture cue is described by an orientation histogram, which is computed based on image derivatives. 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.

The similarity between the model and its candidates is measured by Bhattacharya distance [5].

3 Feature Selection and Model Updating

3.1 Log-Likelihood Ratio Images

To determine the descriptive ability of different features, we compute log-likelihood ratio images [3] [15] based on the histograms of the target and its background. Log-likelihood ratio images are also employed in detecting possible threats to the target.

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 ($p^{(b_{in})}$) is calculated as $\zeta_f^{(b_{in})} = p_f^{(b_{in})}/n_f$ and $\zeta_b^{(b_{in})} = p_b^{(b_{in})}/n_b$, where n_f is the pixel number of the target region and n_b 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.

3.2 Feature Selection

Given m_d features for tracking, the purpose of the feature selection module is to find the best subset feature 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 [3], based on the equality $\text{var}(x) = E[x^2] - (E[x])^2$, the variance of Equation(1) is computed as

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

The variance ratio of the likelihood function is defined as [3]:

$$\text{VR} = \frac{\text{var}(B \cup F)}{\text{var}(F) + \text{var}(B)} = \frac{\text{var}(L; (p_f + p_b)/2)}{\text{var}(L; p_f) + \text{var}(L; p_b)}. \tag{2}$$

3.3 Updating the Target Model

It is necessary to update the target model due to the fact that 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. To reliably update the target model, we propose an approach based on similarities between the initial and current appearance of the target. Similarity θ is measured by a simple correlation based template matching performed between the initial and current frames. The updating is done according to the similarity θ:

$$H_m = (1 - \theta)H_i + \theta H_c, \tag{3}$$

where the H_i is the histogram computed on the initial target; the H_c the histogram of the target current appearance, the H_m the updated histogram of the target.

Template matching is performed between the initial model and the current candidates. Since we do not use the search window that is necessary in template matching-based tracking, the matching process is efficient and brings little computational cost to our algorithm.

In unstable tracking period (When sudden motions or distractions are detected), the classification of the target and background is not reliable. It is difficult to reliably update the target model at these moments. Thus the model is updated when the tracker is in stable states.

4 Motion Estimation and Distraction Detection

4.1 Motion Estimation

The number of particles is adjusted according to motion information of the target. Discriminative mean shift tracking is sufficient to determine the position of a target when it moves smoothly and slowly. More particles are necessary to estimate the correct position of the target when it moves quickly.

We use the efficient motion filters that have been applied in pedestrian detection [16]. We estimate the motion of foreground and background region simultaneously and partially solve the problem brought by dynamic background.

There are five motion filters computed on 5 image pairs:

$$\Delta_i = \frac{1}{n_{Rg}} \int_{\mathbf{x} \in Rg} |I_t(\mathbf{x}) - I_{t+1}^{\tau_i}(\mathbf{x})|, \tag{4}$$

where I_t and I_{t+1} are consequential images, n_{Rg} is the number of pixels in a specific region, and $\tau_i \in \{\diamond, \leftarrow, \rightarrow, \uparrow, \downarrow\}$ which are image shift operators denoting no shift, shift left, shift right, shift up, and shift down for one pixel respectively.

The motion filters are computed on the target and its background region respectively. The results of the last four motion filters ($\Delta_i, i \in \{1, 2, 3, 4\}$) are compared with the absolute differences Δ_0:

$$M_i^f = |\Delta_i^f - \Delta_0^f|, M_i^b = |\Delta_i^b - \Delta_0^b| \tag{5}$$

M_i represent the likelihood that a particular region is moving in a given direction.

We compute the maximum motion likelihood to determine the number of particles for the tracking:

$$M_{max} = \max(|M_i^f - M_i^b|)_{i=1,2,3,4.}. \tag{6}$$

Given the high efficiency of the estimation method, it is performed in each frame before tracking is carried out.

4.2 Distraction Detection

Distractions in the neighborhood of the target have similar appearance to the target. They are possible threats to successful tracking. When the similarity between the target model and its candidate is less than a certain value (ρ^T), distraction detection is performed using spatial reasoning [3] to find peaks besides the target in the log-likelihood ratio images. Note that the log-likelihood ratio images here are back-projection results of the conditional distributions based on selected features.

Assuming that the region R_T actually contains the target and the region R_D is a possible distraction, we want to find the region that have maximum strength of threat to the target. A certain region where the sum of its log-likelihood ratios

has minimum difference with that in the target region is the distraction we want to find:

$$\min(|\sum_{R_D} L^{(b_{in})} - \sum_{R_X} L^{(b_{in})}|) \tag{7}$$

where R_X is a region in the neighborhood of the target.

It is too expensive to compare the sums of log-likelihood ratio in all the possible regions with that in the target region. The searching process can be accelerated using a Gaussian kernel [3]. The value at each pixel in the convolved log-likelihood ratio image with a Gaussian kernel is a weighted sum of the log-likelihood ratios in a circular region surrounding it, normalized by the total weight pixels in that region. First, the log-likelihood image is convolved using a Gaussian kernel. The peak D_T which represents the target region can be found in the convolved image. Second, the target region in the log-likelihood image is removed and the result is convolved using a Gaussian kernel again. The most dangerous distraction is detected by searching for the peak D_D in the convolved image.

The difference between the two peaks represents the threat strength of the distraction:

$$\rho = |D_D - D_T|, \tag{8}$$

The distraction may attract the mean shift tracker to the incorrect position if it is strong enough. We initialize a auxiliary particle set to track the distraction region when ρ is less than the given threshold ρ^T.

5 Auxiliary Particle Filtering

Particle filtering implements recursive Bayesian filter by Monte Carlo simulations. In the implementation, the posterior density is approximated by a weighted particle set $\{s_t^{(n)}, \pi_t^{(n)}\}_{n=1,\cdots,J}$, where $\pi_t^{(n)} = p(\mathbf{z}_t|\mathbf{x}_t = s_t^{(n)})$. We initialize auxiliary particles when sudden motion or distraction are detected. Different strategies are adopted for the generation of particles under these two circumstances.

5.1 Particle Filtering for Sudden Motion

When a sudden motion is detected N_p particles are generated using a stochastic motion model. The number of particles is determined from to the motion computed:

$$J_S = \max(\min(J_0 M_{max}, J_{max}), J_{min}), \tag{9}$$

where J_0 is the coefficient; J_{min} is the smallest number of particles and J_{max} the largest number of particles to maintain reasonable particles.

The motion model is a normal density centered on the previous pose with a constant shift vector:

$$\mathbf{x}_t^j = \mathbf{x}_{t-1} + \mathbf{x}^c + \mathbf{u}_t^j; \tag{10}$$

where $\mathbf{u}_t^j$ is a standard normal random vector and $\mathbf{x}^c$ a constant shift vector from the previous position according to the motion estimation results (it is set to one pixel to the motion direction).

5.2 Particle Filtering for Distraction

After distractions are detected, a joint particle filter with an MRF motion model is initialized [12]. The motion interaction between the target and the distraction $\psi(X_{it}, X_{jt})$ is described by the Gibbs distribution $\psi(X_{it}, X_{jt}) \propto exp(-g(X_{it}, X_{jt})$, where $g(X_{it}, X_{jt})$ is a penalty function approximated by the distance between the target and the distraction.

The posterior on the joint state X_t is approximated as a set of J weighted samples:

$$P(X_t|Z^t) \approx kP(Z_t|X_t) \prod_{ij \in E} \psi(X_{it}, X_{jt}) \sum_J \pi_{t-1}^{(J)} \prod_i P(X_{it}|X_{i(t-1)}^{(J)}),$$

where the samples are drawn from the joint proposal distribution; k is a normalizing constant that does not depend on the state variables; E is edges in the MRF model; the samples are weighted according to the factored likelihood function:

$$\pi_t^{(s)} = \prod_i^2 P(Z_{it}|X_{it}^{(s)}) \prod_{ij \in E} \psi(X_{it}^{(s)}, X_{jt}^{(s)}).$$

where Z_{it} are measurement nodes.

5.3 Algorithm Summary

In summary, the detailed steps of the proposed tracking algorithm are:

Algorithm: *Discriminative Mean-Shift Tracking with Auxiliary Particles*

Input: t video frames $I_1, \ldots, I_t$;
Initial target region given in the first frame I_1
Output: target regions in $I_2, \ldots, I_t$
Initialization in I_1

1. **Save** the initial target appearance for model updating;
2. **Compute** the similarity (S_1) between the target model and the candidate.

For each new frame I_j:
Estimate the motion (M_j) on the consequential frames;
IF $M_j > M_T$
THEN initialize particles according to the motion estimated.
ELSE
IF the similarity is less than a given threshold ($S_{j-1} < S^T$)
THEN detect distractions in the neighborhood of the target
If Distraction is detected ($\rho < \rho^T$)
Initialize MRF particles;
Else
Update the target model.
End If
End If
End If
Estimate the position of the target.
Compute the similarity S_j for next frame.
End For

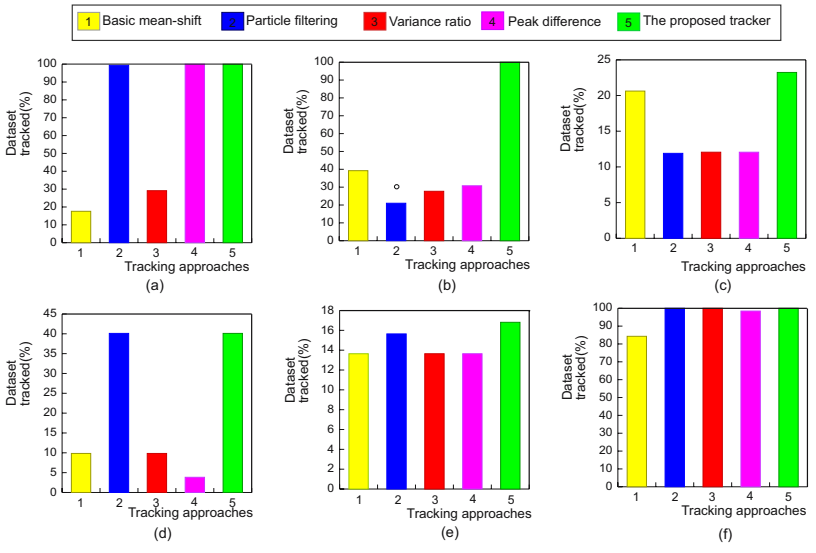

Fig. 1. Tracking results using different tracking approaches. Tests are performed on (a) EgTest01; (b) EgTest02; (c) EgTest03; (d) EgTest04; (e) EgTest05; and (f) Redteam.

6 Experimental Results

To illustrate the performance of the proposed tracker, we have implemented and tested it on a wide variety of challenging image sequences in different environments and applications. Due to space limitation, we only show the results on the public CMU datasets with ground truth [4]. The datasets include 6 sequences: EgTest01, EgTest02, EgTest03, EgTest04, EgTest05 and Redteam. There are different factors that make the tracking challenging: different viewpoints (these sequences are captured by moving cameras); similar objects nearby; sudden motions; illumination changes; reflectance variations of the targets; and partial occlusions.

The tracking results are compared with the basic mean shift and particle filtering trackers. Since the proposed tracker updates the target model based on feature selection, it is reasonable to compare it with those adaptive trackers. The variance ratio and peak difference [3] trackers are included for this purpose. In the particle filtering tracker, the target model is represented by $12 \times 12 \times 12$-bin RGB histograms. There are 100 samples in the sample set. RGB histograms are also adopted in the basic mean shift algorithm. The similarity measure is Bhattacharya distance between the model and its candidate.

The most important criterion for the comparison is the percentage of dataset tracked, which is the number of the 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 tracking success rates achieved by each tracker are compared and the results are shown in Fig. 1. The proposed tracker gives the

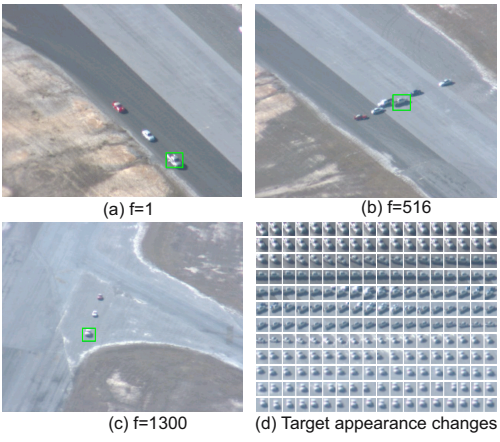

Fig. 2. Tracking results of the EgTest02 sequence

best results (or has same results with another tracker) in all the test sequences. Those comparisons demonstrate that the proposed tracking algorithm has better performance than other trackers. In Fig. 2, the tracking results for EgTest02 are shown. Despite the distractions and sudden motions in the sequence, the proposed tracker completes the tracking successfully. Fig. 2.(d) illustrates how the appearance of the target changes over time.

There are sudden motions and image blur in the EgTest04, which leads to the failure of the basic mean-shift tracker. The proposed tracker detects this motion successfully and initializes auxiliary particles. These particles help the proposed tracker to conquer the problem brought by the sudden motion.

The running time of the proposed tracker depends on the difficulty level of the image sequence being tracked. If sudden motions or distractions happen frequently, its efficiency is low. Otherwise it has high efficiency because the mean shift algorithm is adopted in most cases. The current implementation ran 16 frames per second (average speed) on a Intel Centrino 1.6GHz laptop with 1G RAM when applied to images of size 640×480. The average running time includes time to do the main tracking algorithm, to read image file from a USB disk, and to display color images with the object bounding box overlaid.

7 Conclusion and Future Work

We describe a discriminative mean shift tracking algorithm with auxiliary particles in the pursuit of robust and efficient tracking. The arrangement of the particle filtering and the mean shift algorithm is based on the difficulty of the tracking which is indicated by sudden motions and distractions. The model updating strategy in our tracker can effectively deal with appearance changes of targets. The proposed approach provides better performance than those of the mean shift, particle filtering and other trackers.

We are going to investigate how to extend the proposed method to multi-target tracking, in which multiple mean shift searching is necessary.

References

1. Bradski, G.R.: 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)
2. Cai, Y., Freitas, N., Little, J.: Robust Visual Tracking for Multiple Targets. In: Little, J. (ed.) Proc. of 6th Europearn Conf. on Computer Vision, pp. 893–908 (2006)
3. Collins, R.T., Liu, Y.: On-line Selection of Discriminative Tracking Features. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(10), 1631–1643 (2005)
4. Collins, R.T., Zhou, X., Teh, S.K.: An Open Source Tracking Testbed and Evaluation Web Site. In: PETS 2005. IEEE Int'l Workshop on Performance Evaluation of Tracking and Surveillance (January 2005)
5. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based Object Tracking. IEEE Trans. Pattern Analysis Machine Intelligence 25(5), 564–577 (2003)
6. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley and Sons Press, Chichester (1991)
7. Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/non-Gaussain Bayesian state estimation. IEEE Proc. 140(2), 107–113 (1993)
8. Gevers, T., Smeulders, A.W.M.: Color based object recognition. Pattern Recognition 32(3), 453–464 (1999)
9. Isard, M., Blake, A.: Condensation - conditional density propagation for tracking. Int'l Journal of Computer Vision 29(1), 2–28 (1998)
10. Isard, M., Blake, A.: ICONDENSATION: unifying low-level and high-level tracking in a stochastic framework. In: Proc. of 5th Europearn Conf. on Computer Vision, vol. I, pp. 893–908 (1998)
11. Jhne, B., Scharr, H., Krkel, S.: Handbook of Computer Vision and Applications. In: Jhne, B., Hauecker, H., Geiler, P. (eds.), vol. 2, pp. 125–151. Academic Press, London (1999)
12. Khan, Z., Balch, T., Dellaert, F.: An MCMC-based particle filter for tracking multiple interacting targets. In: Proc. of 5th Europearn Conf. on Computer Vision, vol. I, pp. 893–908 (2004)
13. Shan, C., Tan, T., Wei, Y.: Real-time hand tracking using a mean shift embedded particle filter. Pattern Recognition 40(7), 1958–1970 (2007)
14. Sullivan, J., Rittscher, J.: Guiding Random Particles by Deterministic Search. In: Proc. of Eighth IEEE Int'l Conf. on Computer Vision, vol. I, pp. 323–330 (2001)
15. Swain, M., Ballard, D.: Color Indexing. Int'l Journal of Computer Vision 7, 11–32 (1991)
16. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. Int'l Journal of Computer Vision 63(2), 153–161 (2005)
17. Wang, J., Yagi, Y.: Integrating Shape and Color Features for Adaptive Real-time Object Tracking. In: IEEE Int'l Conf. on Robotics and Biomimetics 2006 (2006)