Mean-Shift Tracking with Random Sampling

Alex Po Leung, Shaogang Gong

Department of Computer Science
Queen Mary, University of London, London, E1 4NS

Abstract

In this work, boosting the efficiency of Mean-Shift Tracking using random sampling is proposed. We obtained the surprising result that mean-shift tracking requires only very few samples. Our experiments demonstrate that robust tracking can be achieved with as few as even 5 random samples from the image of the object. As the computational complexity is considerably reduced and becomes independent of object size, the processor can be used to handle other processing tasks while tracking. It is demonstrated that random sampling significantly reduces the processing time by two orders of magnitude for typical object sizes. Additionally, with random sampling, we propose a new optimal on-line feature selection algorithm for object tracking which maximizes a similarity measure for the weights of the RGB channels. It selects the weights of the RGB channels which discriminate the object and the background the most using Steepest Descent. Moreover, the spatial distribution of pixels representing the object is estimated for spatial weighting. Arbitrary spatial weighting is incorporated into Mean-Shift Tracking to represent objects with arbitrary or changing shapes by picking up non-uniform random samples. Experimental results demonstrate that our tracker with online feature selection and arbitrary spatial weighting outperforms the original mean-shift tracker with improved computational efficiency and tracking accuracy.

1 Introduction

Much effort has been made to solve the problem of real-time object tracking over the years. However, tracking algorithms still suffer from fundamental problems including drifts away from targets [3] (partially due to change of viewpoint), inability to adapt to changes of object appearance, dependence on the first frame for template matching [4], instability to track objects under deformations (e.g. deformed contours), the inefficiency of Monte Carlo simulations for temporal tracking [5], and reliance on gradients by active contours [6], i.e. problems with similar intensities on the background and the object, or high gradient edges on the object itself. These problems are due to the complexity of the object dynamics. We also have to deal with difficult tracking conditions which include illumination changes, occlusions, changes of viewpoint, moving cameras and non-translational object motions like zooming and rotation.

Recently, Mean-Shift Tracking [2] has attracted much attention because of its efficiency and robustness to track non-rigid objects with partial occlusions, significant clutter

and variations of object scale. As pointed out by Yang and Duraiswami [7], the computational complexity of traditional Mean-Shift Tracking is quadratic in the number of samples, making real-time performance difficult. Although the complexity can be made linear with the application of a recently proposed fast Gauss transform [7], tracking in real-time remains a problem when large or multiple objects are involved. We propose to boost the efficiency of mean-shift tracking using random sampling. When the computational efficiency for mean-shift tracking with random sampling was evaluated, we obtained the surprising result that mean-shift tracking requires only very few samples. We show that robust tracking can be achieved with as few as even 5 random samples from the image of the object. With random sampling, the computational complexity of Mean-Shift Tracking is independent of object size. Large or multiple objects can be tracked in real time. As the computational complexity is considerably reduced, the processor can be used to handle other processing tasks while tracking. Near real-time performance is obtained even in our Matlab implementation which demonstrates that Mean-Shift Tracking with random sampling runs much faster than 30 frames per second as a Matlab implementation is typically at least two orders of magnitude slower than an implementation with C. It is also shown that, instead of passing hundreds of samples to a traditional mean-shift tracker, only 5 random samples are required for the mean-shift tracker to track objects with a relatively simple distribution and 15 samples for a typical distribution. In our experiments, random sampling significantly reduces the processing time by two orders of magnitude for typical object sizes.

In addition, with random sampling, we propose a new optimal on-line feature selection algorithm for object tracking which maximizes a similarity measure for the weights of the RGB channels. It selects the weights of the RGB channels which discriminate the object and the background the most using Steepest Descent. However, the problem is that the Bhattacharyya coefficient as the objective function of the weights for the RGB channels is not uni-modal. A simpler measure using random sampling is proposed so that Steepest Descent can be applied to find the optimal weights.

Moreover, the spatial distribution of pixels representing the object is estimated for spatial weighting. Arbitrary spatial weighting is incorporated into Mean-Shift Tracking to represent objects with arbitrary or changing shapes by picking up non-uniform random samples. For Mean-Shift Tracking, the probability of the color is derived using a convex and monotonic decreasing kernel profile with a smaller weight to the pixels farther from the centroid of the object to increase the robustness of the tracking. The multivariate Epanechnikov kernel [2] and the Gaussian kernel [7] have been successfully applied to Mean-Shift Tracking. However, they are not able to deal with objects in arbitrary or changing shapes very well. Apart from using a convex and monotonic decreasing kernel profile to model the reliabilities of different parts of the object, pixels representing the object are used to estimate the spatial importance or weighting in each part of the tracking subwindow. The pixels of the object are extracted by segmentation using Normalized Cut [8]. Random samples are picked up from the candidate and model images according to our estimate. Instead of using all samples from the candidate and model images, the random samples are given to the mean-shift tracker.

2 Computational Efficiency of Mean-Shift Tracking with Uniform Random Sampling

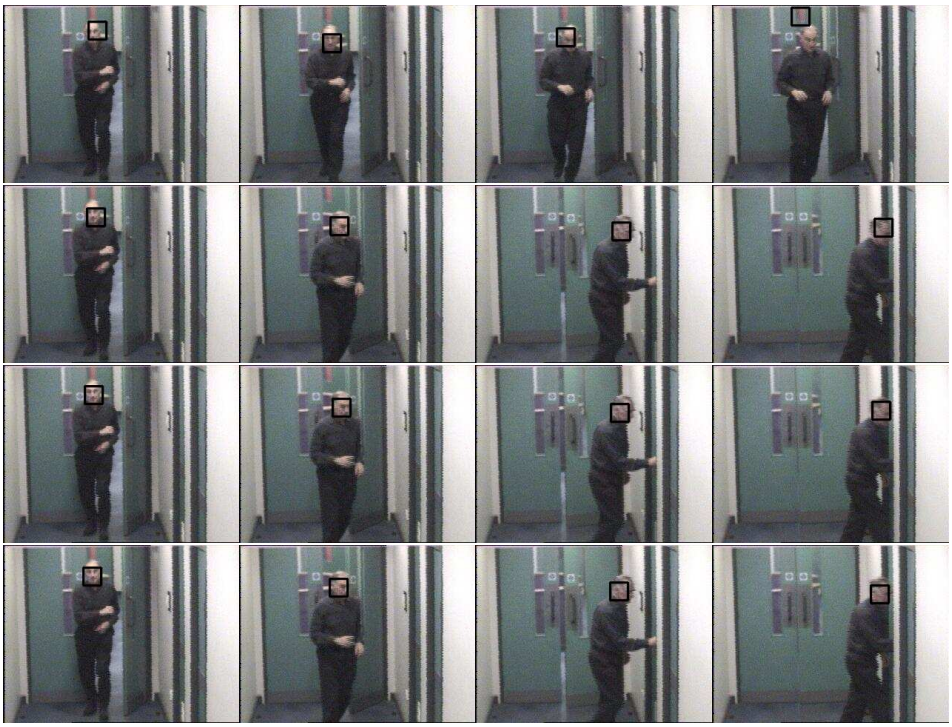

Figure 1: Experiments 1, 2, 3 and 4: The four rows of images represent four separate video sequences produced by four experiments respectively. In the first experiment (first row), only 5 random samples are picked from each of the candidate and model images, 10 samples in the second experiment (second row), 15 samples in the third (third row) and 150 samples in the fourth (fourth row).

In this section, we propose to boost the efficiency of mean-shift tracking using random sampling and evaluate the efficiency of the proposed method. When the computational efficiency for mean-shift tracking with random sampling was evaluated, we obtained the surprising result that mean-shift tracking requires only very few samples. We show that robust tracking can be achieved with as few as even 5 random samples from the image of the object. With random sampling, the computational complexity of Mean-Shift Tracking is independent of object size. Near real-time performance is obtained even in our Matlab implementation because, instead of passing hundreds of samples to a traditional mean-shift tracker, only 5 random samples are required for the mean-shift tracker to track objects with a relatively simple distribution and 15 samples for a typical distribution. The speed of our tracker is 2.17 fps (frames per second) to track the head of a person in a given video sequence while the speed of a traditional implementation used in [7] is 0.011 fps (197 times slower) with the same tracking sub-window size of 24x25 (600 samples without random sampling).

Figure 2: Experiments 5, 6, 7 and 8: The four rows of images represent four separate video sequences produced by four experiments respectively. In Experiment 5 (first row), only 2 random samples are picked from each of the candidate and model images, 3 samples in Experiment 6 (second row), 5 samples in Experiment 7 (third row) and all samples from the candidate and model images (traditional Mean-Shift Tracking) in Experiment 8 (fourth row).

Our first four experiments, as shown in Figure 1, evaluate the number of random samples required to track a typical object which is a human face in the experiments. In the first experiment, only 5 random samples are picked from each of the candidate and model images, 10 samples in the second experiment, 15 samples in the third and all samples from the candidate and model images (traditional Mean-Shift Tracking) in the fourth. Tracking fails with too few samples. The tracker fails with 5 samples from the image of the object. With 10 samples, as shown in the second row of Figure 1, the tracker tracks the object successfully but the trajectory is not very stable when compared with the tracker with 150 samples (fourth row). There is no difference between the tracking performance of the mean-shift tracker with 15 samples (third row) and that of the tracker with 150 samples. A larger number of samples more than 15 does not make any difference to the tracking performance of the tracker. On Matlab, the time required for the tracking in Experiments 1, 2, 3 and 4 are 2.95 fps (frames per second), 2.63 fps, 2.18 fps and 0.04 fps respectively. Experiments 5, 6, 7 and 8, as shown in Figure 2, evaluate the number of random samples required to track an object with a relatively simple distribution which is the head of a person. In Experiment 5, only 2 random samples are picked from each of the candidate and model images, 3 samples in Experiment 6, 15 samples in Experiment 7

and all samples from the candidate and model images (traditional Mean-Shift Tracking) in Experiment 8. The tracker fails with 2 samples from the image of the object. With 3 samples, as shown in the second row of Figure 2, the tracker tracks the object successfully but the trajectory is not very stable when compared with the original mean-shift tracker (fourth row). There is no difference between the tracking performance of the mean-shift tracker with 5 samples (third row) and that of traditional Mean-Shift tracking. Therefore, for an object with a relatively simple color distribution, a larger number of samples more than 5 does not make any difference to the tracking performance of the tracker. With our Matlab implementation, the time required for the tracking in Experiments 5, 6, 7 and 8 are 3.19 fps (frames per second), 3.12 fps, 2.17 fps and 0.011 fps respectively. Successful tracking with 5 random samples is 197 times faster than traditional Mean-Shift Tracking with the same tracking sub-window size of 24x25 (600 samples without random sampling). The computational complexity of traditional Mean-Shift Tracking is quadratic in the number of samples. In our experiments, random sampling significantly reduces the processing time by two orders of magnitude for typical object sizes.

2.1 Optimal On-Line Feature Selection with Random Sampling

An on-line feature selection method for Mean-Shift Tracking is proposed recently [1]. It is demonstrated that Mean-Shift Tracking can be made adaptive to the changing environment and more robust with the method. It adapts to changing appearances of both tracked object and scene background by selecting the most discriminative feature with discrete weighting for the RGB pixel values. The weight for each of the RGB pixel values can be set to either -2, -1, 0, 1 or 2. Hence, the most discriminative feature is a linear combination of the RGB pixel values. Because of redundancy, a pool of only 49 candidate features are left for feature selection. All features are, then, ranked and the most discriminative feature is selected accordingly. However, because of the nature of the discrete weighting, the method prevents the feature selection from being optimal. We propose an optimal on-line feature selection method for mean-shift tracking using Steepest Descent for our real-valued weights, i.e. $\alpha_i \in \mathbb{R}, i = 1,2,3$.

The objective of our on-line feature selection for object tracking is to maximize a similarity measure for the weights of the RGB channels. It selects the weights of the RGB channels which discriminate the object and the background the most. However, the problem is that the Bhattacharyya coefficient as the objective function of the weights for the RGB channels is not uni-modal. Local optimization techniques could not be applied to selecting the best features whereas global optimizations are undesirable because of its complexity. A simpler measure using random sampling is proposed so that Steepest Descent can be applied to find the optimal weights. For mean-shift tracking, traditionally, the probabilities of the color u in the target model and the target candidate are given by

$$\hat{q}_u = C\sum_{i=1}^{n} k(||x_i^*||^2)\delta[b(x_i^*) - u], \text{ and} \tag{1}$$

$$\hat{p}_u(y) = C_h\sum_{i=1}^{n_h} k(||\frac{y - x_i}{h}||^2)\delta[b(x_i) - u] \tag{2}$$

where C and C_h are normalization factors, x_i^* and x_i are the pixel locations of the target model and the target candidate, and k is a convex and monotonic decreasing kernel profile.

The distance between the two discrete distributions is defined as [2]

$$d(y) = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]} \tag{3}$$

where

$$\hat{\rho}(y) = \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u}, \tag{4}$$

the sample estimate of the Bhattacharyya coefficient between p and q.

Instead of maximizing the Bhattacharyya coefficient with the multivariate Epanechnikov kernel for the estimates $\hat{p}$ and $\hat{q}$, we pick up random samples from the object and the background distributions. Non-uniform random sampling can be used to select samples from the model image for spatial weighting. The chosen random samples from the two distributions are, then, compared. Motivated by the χ^2 difference, our similarity measure based on the sum of squared difference of the random samples is defined to be

$$g = \sum_j (\sum_i \alpha_i f_{ij}^* - \sum_i \alpha_i f_{ij})^2 \tag{5}$$

with the constraint

$$\sum_i \alpha_i^2 = 1 \tag{6}$$

where f_{ij}^* and f_{ij} are the pixel values of Sample j for Channel i from the target model and the target candidate respectively. α_i^2 are the weights for the RGB channels with $i = 1, 2, 3$ representing R, G and B. The first and second derivatives of the function g are

$$\frac{\partial g}{\partial \alpha_l} = \sum_j 2(\sum_i \alpha_i f_{ij}^* - \sum_i \alpha_i f_{ij})(f_{lj}^* - f_{lj}), \text{ and} \tag{7}$$

$$\frac{\partial^2 g}{\partial \alpha_l^2} = \sum_j 2(f_{lj}^* - f_{lj})^2. \tag{8}$$

Notice that $g(\alpha)$ is a convex function. We should, thus, maximize g in polar coordinates instead of maximizing $g(\alpha)$. In polar coordinates,

$$\alpha_1 = r \sin\phi \cos\theta, \tag{9}$$

$$\alpha_2 = r \sin\phi \sin\theta, \text{ and} \tag{10}$$

$$\alpha_3 = r \cos\phi. \tag{11}$$

By combining Equations 6, 9, 10 and 11, we obtain $r = 1$. Therefore,

$$g = \sum_j (\sin\phi \cos\theta (f_{1j}^* - f_{1j}) + \sin\phi \sin\theta (f_{2j}^* - f_{2j}) + \cos\phi (f_{3j}^* - f_{3j}))^2. \tag{12}$$

The partial derivatives of g are

$$\frac{\partial g}{\partial \phi} = \sum_j 2[\sin\phi \cos\theta (f_{1j}^* - f_{1j}) + \sin\phi \sin\theta (f_{2j}^* - f_{2j}) + \cos\phi (f_{3j}^* - f_{3j})]$$
$$[\cos\phi \cos\theta (f_{1j}^* - f_{1j}) + \cos\phi \sin\theta (f_{2j}^* - f_{2j}) - \sin\phi (f_{3j}^* - f_{3j})], \text{ and} \tag{13}$$

$$\frac{\partial g}{\partial \theta} = \sum_j 2[\sin\phi \cos\theta (f_{1j}^* - f_{1j}) + \sin\phi \sin\theta (f_{2j}^* - f_{2j}) + \cos\phi (f_{3j}^* - f_{3j})]$$
$$[-\sin\phi \sin\theta (f_{1j}^* - f_{1j}) + \sin\phi \cos\theta (f_{2j}^* - f_{2j})]. \tag{14}$$

2.2 Tracking Objects in Arbitrary Shapes with Non-Uniform Random Sampling

1. Use Normalized Cut to extract the pixels of the object in the user-selected subwindow of the first frame.
2. Estimate the spatial importance, $o(\mathbf{x})$, using the spatial distribution of the pixels representing the object with a histogram.
3. Select random samples, x_i from the target model according to our estimate , $\hat{o}(\mathbf{x})$, using the rejection method.
4. Select random samples, x_i^* from the target candidate according to our estimate , $\hat{o}(\mathbf{x})$, using the rejection method.
5. Use the random samples for Mean-Shift Tracking instead of processing all samples from the candidate and model images.

Table 1: The algorithm to Track Objects in Arbitrary Shapes with Non-Unform Sampling

For Mean-Shift Tracking, the probability of the color is derived using a convex and monotonic decreasing kernel profile with a smaller weight to the pixels farther from the centroid of the object to increase the robustness of the tracking. The multivariate Epanechnikov kernel [2] and the Gaussian kernel [7] have been successfully applied to Mean-Shift Tracking. However, they are not able to deal with objects in arbitrary or changing shapes very well. Apart from using a convex and monotonic decreasing kernel profile to model the reliabilities of different parts of the object, pixels representing the object are used to estimate the spatial importance or weighting in each part of the tracking subwindow. We define the spatial importance to be the probability that Pixel $\mathbf{x}$ is part of the object, $o(\mathbf{x})$. Instead of estimating $o(\mathbf{x})$ with a number of the examples of the object, only one sample, the original model image, is used. This avoids the problem for the user to collect a number of examples similar in appearance to the object. It is demonstrated in our experiments that our estimate using only one single sample is very effective.

The spatial importance, $o(\mathbf{x})$, can be estimated using the spatial distribution of the pixels representing the object. As the distribution is two-dimensional, a histogram would suffice for the purpose. Therefore, our estimate, $\hat{o}(\mathbf{x})$, is the two-dimensional spatial histogram of the pixels corresponding to the object. The pixels of the object are extracted by segmentation using Normalized Cut [8]. Random samples are picked up from the candidate and model images according to our estimate , $\hat{o}(\mathbf{x})$, for the two-dimensional spatial distribution representing the spatial importance using the rejection method. Furthermore, instead of using all samples from the candidate and model images, the random samples are given to the mean-shift tracker. Our algorithm for tracking objects in arbitrary shapes with non-unform sampling is summarized in Table 1.

3 Experimental Results

Our further experiments investigate the performance of the mean-shift tracker with our methods for online feature selection and arbitrary spatial weighting. It is demonstrated that our tracker with online feature selection and arbitrary spatial weighting outperforms the original mean-shift tracker with improved computational efficiency and tracking accuracy. In Experiment 9, the black hat of a person against a red wall and a glass door as the background is tracked (Figure 3). It is shown that the tracker combined with our

Figure 3: In Experiment 9 (first two rows), the black hat of a person against a red wall and a glass door as the background is tracked. It is shown that the tracker combined with our on-line feature selection method tracks the black hat successfully when the hat moves in front of a glass door outside a dark corridor. In the bottom row, Experiment 9 is repeated without on-line feature selection. The original mean-shift tracker is distracted by the glass door on the background.

Figure 4: The face of a person against a red wall is tracked successfully with our on-line feature selection method in Experiment 10.

on-line feature selection method tracks the black hat successfully when the hat moves in front of a glass door outside a dark corridor. Moreover, in Experiment 10, the face of a person against a red wall is tracked successfully with our on-line feature selection method as shown in Figure 4. On Matlab, our on-line feature selection algorithm using Steepest Descent on average converges in 0.4 second for each frame and the average number of iterations for convergence is 23. Figure 5 are two plots showing the weights for the RGB channels, α_1, α_2 and α_3. The plot on the top shows the change of the weights for the RGB channels, α_1, α_2 and α_3, when the hat of the person is tracked in Experiment 9 and the plot on the bottom shows the change of α_1, α_2 and α_3 when the face of the person is tracked in Experiment 10. From Frame 35 to Frame 60, the hat and the face are tracked against the red wall. The red channel is considered more important by the tracker and, thus, given the highest weight. From Frame 210 to Frame 270, the fluctuation of the

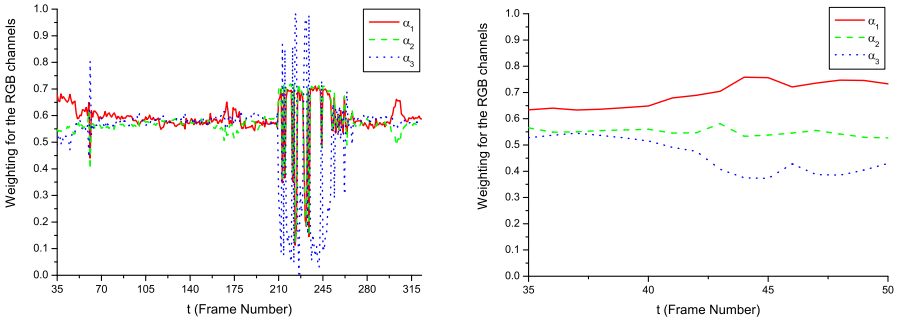

Figure 5: The plot on the top shows the weights for the RGB channels, α_1, α_2 and α_3, when the hat of a person is tracked in Experiment 9 and the plot on the bottom shows α_1, α_2 and α_3 when the face of a person is tracked in Experiment 10. From Frame 35 to Frame 60, the hat and the face are tracked against the red wall. The red channel is considered more important by the tracker and, thus, given the highest weight. From Frame 210 to Frame 270, the fluctuation of the weights is triggered by the white frame of the glass door.

Figure 6: A car on a busy road is tracked in Experiment 11 and a pedestrian is tracked in Experiment 12. It is shown that the tracker combined with non-uniform sampling for arbitrary spatial weighting tracks the car and the pedestrian successfully.

weights is triggered by the white frame of the glass door. To evaluate our method using non-uniform sampling for arbitrary spatial weighting, a car on a busy road is tracked in Experiment 11 and a pedestrian is tracked in Experiment 12 (Figure 6). It is shown that the tracker combined with non-uniform sampling for arbitrary spatial weighting tracks the car and the pedestrian successfully. The spatial distributions of the pixels from the car and the pedestrian (Figure 7) extracted by Normalized Cut are used to build two-dimensional histograms for the estimate of the spatial importance, $\hat{o}(\mathbf{x})$. The number of bins used is 9 for the two-dimensional histograms in both of the experiments. Furthermore, Experiment 9 is repeated without on-line feature selection. Figure 3 shows that the original mean-

shift tracker is distracted by the glass door on the background. Experiment 12 is repeated without non-uniform sampling for arbitrary spatial weighting. As shown in Figure 7, the original mean-shift tracker fails to track the pedestrian.

4 Conclusion

To conclude, we have proposed Mean-Shift Tracking with random sampling which is shown to reduce the processing time by two orders of magnitude for typical object sizes. Besides, a new optimal on-line feature selection algorithm for object tracking has been proposed to maximize a similarity measure for the weights of the RGB channels. It selects the weights of the RGB channels which discriminate the object and the background the most using Steepest Descent. Finally, arbitrary spatial weighting is incorporated into Mean-Shift Tracking to represent objects with arbitrary or changing shapes by picking up non-uniform random samples. Our experimental results demonstrated that our tracker with online feature selection and arbitrary spatial weighting outperforms the original mean-shift tracker with improved computational efficiency and tracking accuracy.

References

[1] Collins, R. T., Y. Liu, On-Line Selection of Discriminative Tracking Features, *Proceedings of the IEEE International Conference on Computer Vision (ICCV'03)*, October 2003.

[2] D. Comaniciu, , V. Ramesh and P. Meer, Kernel-Based Object Tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, May 2003.

[3] I. Matthews, T. Ishikawa and S. Baker, "The Template Update Problem," *PAMI(26)*, No. 6, 2004, pp. 810-815.

[4] F. Jurie and M. Dhome, "Hyperplane Approximation for Template Matching," *IEEE PAMI 24(7)*, 996-1000, 2002.

[5] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *Transaction of Signal Processing*, 50(2):174-188, 2002.

[6] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *IJCV*, 29, 1, 5–28, (1998).

[7] Changjiang Yang, Ramani Duraiswami, Larry S. Davis, "Efficient Mean-Shift Tracking via a New Similarity Measure," *CVPR*, 1, 176-183, 2005.

[8] J. Shi and J. Malik, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

Figure 7: In the left image, Experiment 12 is repeated without non-uniform sampling for arbitrary spatial weighting. The original mean-shift tracker fails to track the pedestrian. The right image shows that the spatial distributions of the pixels from the objects (Experiment 11 and Experiment 12) extracted by Normalized Cut are used to build 2-dimensional histograms for the estimate of the spatial importance, $\hat{o}(\mathbf{x})$.