Orientation and Scale Invariant Mean Shift Using Object Mask-Based Kernel

Kwang Moo Yi   Ho Seok Ahn   Jin Young Choi
EECS Department, Automation and System Research Institute (ASRI)
Seoul National University, Seoul Korea,
E-mail: {kmyi, hsahn}@neuro.snu.ac.kr, jychoi@snu.ac.kr

Abstract

In this paper, we propose a new method for object tracking based on mean shift algorithm using a kernel which has the shape of the target object, and with probabilistic estimation of the orientation change and scale adaptation. The proposed method uses an object mask to construct a kernel which has the shape of the actual object for tracking. Orientation is adjusted using probabilistic estimation of orientation and scale is adapted using a newly proposed descriptor for scale. Tests results show that the proposed method is robust to background clutter and tracks objects very accurately.

1. Introduction

Tracking of objects using the mean shift algorithm is a popular way in the field of object tracking. The algorithm has its advantages in the fact that it is relatively easy to implement and it does not require heavy computation, and yet it shows robust results in practical object tracking tasks. However, the traditional mean shift algorithm has three major problems. The first problem is that although choosing the correct scale (i.e. the correct kernel bandwidth) is critical to the tracking performance, there is no robust way of adapting to scale changes. The second problem is adapting to orientation changes. The last problem is the inclusion of background information inside the object model due to the restriction of kernel shape. These problems greatly affect the tracking performance and need to be solved.

The first problem was intuitively solved in [1] by the 10% method. This method however, does not work well due to its nature of preferring the smaller kernel. To overcome this problem, R. Collins proposed a method using difference of Gaussian mean shift kernel in scale space [2]. This method performs well but needs heavy computation and is not suitable for real-time purposes. The second problem has not been covered much due to the limitation of the shape of kernels (which is the third problem), but still this problem also affects the tracking performance. The third problem was usually approached with the use of anisotropic kernels such as ellipsoids [1]. This method still contains background pixels inside which causes background clutter problems.

Our newly proposed algorithm overcomes the three problems stated above by adapting to scale using a newly defined descriptor for scale, estimating the orientation using histograms constructed for each orientation of the object model, and using a kernel constructed based on a mask for the object model. And also, our method does not need additional iteration for orientation and scale so the fast computation time of the mean shift algorithm is preserved. The test results show that the proposed algorithm is superior to the traditional mean shift and is also comparable to another popular tracking algorithm, with high computation, the particle filter, as in [3], [4], and [5].

2. Mean Shift Tracking: Brief Review

In this section, since our method is based on the traditional mean shift method, we give a brief review of the traditional mean shift algorithm. The mean shift method is a fast way of finding the local maxima of a sample distribution iteratively from a given starting position. In the field of object tracking this sample is the color observed at a pixel $\mathbf{x}$. To this $\mathbf{x}$, the sample weight $w(\mathbf{x})$ is defined as the following:

$$w(\mathbf{x}) = \sqrt{h_m(I(\mathbf{x})) / h_c(I(\mathbf{x}))}\ , \qquad (1)$$

where $I(\mathbf{x})$ is the pixel color, h_m and h_c being the color distribution functions generated from the model

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

and candidate object regions, respectively. If we let the initial hypothesized position be $\hat{\mathbf{x}}_{old}$, the computed new position be $\hat{\mathbf{x}}_{new}$, $\Delta\mathbf{x} = \hat{\mathbf{x}}_{new} - \hat{\mathbf{x}}_{old}$, and $K(.)$ to be the radially symmetric kernel defining the tracking object region respectively, then using the weight above, the mean shift vector is computed as the following:

$$\Delta\mathbf{x} = \frac{\sum_i K(\mathbf{x}_i - \hat{\mathbf{x}}_{old}) w(\mathbf{x}_i)(\mathbf{x}_i - \hat{\mathbf{x}}_{old})}{\sum_i K(\mathbf{x}_i - \hat{\mathbf{x}}_{old}) w(\mathbf{x}_i)} . \qquad (2)$$

This mean shift vector is an estimated of the gradient of the sample distribution and using this mean shift vector, tracking of the object is performed iteratively.

3. The Proposed Method

The following subsections describe the kernel and methods the proposed method uses.

3.1. The Proposed Kernel

In traditional mean shift, a symmetric kernel or an isotropic kernel is used for tracking. These kernels are not capable of describing the object's shape accurately. Thus, these kernels, regardless of their choice, have background information inside the object model. This brings background clutter problems and may result in tracking failures. To overcome these problems, we propose a new kernel which is not symmetric or isotropic in shape.

In a lot of cases, tracking is done after detection. From the detection results, a mask of the foreground object can be created. Using these kinds of masks, we create a kernel which the value of the kernel is the distance from the boundary of the mask (illustrated in figure 1.).

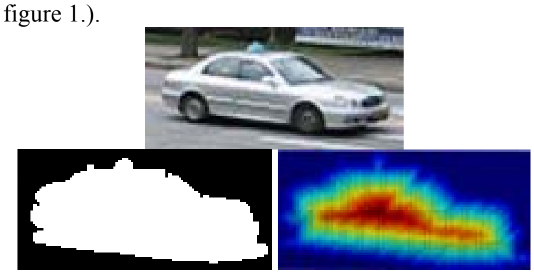


Figure 1. The original image (top), the detected image mask (bottom left) and the constructed kernel (bottom right).

This kernel is a smooth kernel and can be used efficiently for mean shift. Using this kernel allows the minimum inclusion of the background information inside the object model and thus makes the tracking algorithm robust to background clutters. This kernel is similar to the one proposed by A. Yilmaz in [6] but differs in the fact that it is based on object mask which can be given by detection systems beforehand.

3.2. Orientation Estimation

When we do not consider scale change, at the last steps of the iteration when the tracking of the object's translation is almost finished, most of the object is likely to be inside the tracking window, and also since the time difference is very small between frames, the object's orientation is likely to change little. This allows the assumption that the target candidate's color distribution has not changed much from the object model in the last steps of the mean shift iteration. If we let p_m and p_c be the probability with respect to the object model and target candidate respectively, under this assumption we can assume the following:

$$\hat{p}_c(\Delta\theta \mid \upsilon) \approx p_m(\Delta\theta \mid \upsilon) , \qquad (3)$$

where $\Delta\theta$ is the relative angle of the object inside the tracking window, υ is the color value, and ^ denotes the estimator, respectively. From this approximation, we can derive the following:

$$\begin{aligned} \hat{p}_c(\Delta\theta) &= \sum_\upsilon \hat{p}_c(\Delta\theta \mid \upsilon) p_c(\upsilon) \\ &\approx \sum_\upsilon p_m(\Delta\theta \mid \upsilon) p_c(\upsilon), \end{aligned} \qquad (4)$$

where $p_m(\Delta\theta \mid \upsilon)$, which is the probability distribution for orientation according to each color of the object model, can be constructed as the following:

$$\begin{aligned} p_m(\Delta\theta \mid \upsilon) &= \frac{p_m(\Delta\theta, \upsilon)}{p_m(\upsilon)} \\ &= \frac{p_m(\upsilon \mid \Delta\theta) p_m(\Delta\theta)}{\sum_{\Delta\theta} p_m(\upsilon \mid \Delta\theta) p_m(\Delta\theta)} . \end{aligned} \qquad (5)$$

The probability of color values for each $\Delta\theta$, $p_m(\upsilon \mid \Delta\theta)$ can be calculated using histograms constructed for each orientation range. From equations (3), (4), (5) we can obtain the $\hat{E}(\Delta\theta)$ estimator for the expectation of $\Delta\theta$:

$$\hat{E}(\Delta\theta) = \sum_{\Delta\theta} \Delta\theta \hat{p}_c(\Delta\theta) . \qquad (6)$$

For objects with symmetric shape, the probability of orientation can be similar for both positive and negative angle directions (i.e. $\hat{E}(\Delta\theta) \simeq \hat{E}(-\Delta\theta)$). Therefore, we estimate the direction of the orientation in another way. We divide the tracking window into four divisions and investigate the average of the sample weights (from equation (1)) inside these areas. If we denote the average of the weights of the top-left, top-right, bottom-left, bottom-right regions $w_{\mathrm{avg,tl}}$, $w_{\mathrm{avg,tr}}$, $w_{\mathrm{avg,bl}}$, and $w_{\mathrm{avg,br}}$ respectively, and denote the direction of the orientation as δ , we can define the value of δ as +1 if $\max(w_{\mathrm{avg,tl}} , w_{\mathrm{avg,br}}) > \max(w_{\mathrm{avg,tr}} , w_{\mathrm{avg,bl}})$ and as –1 if otherwise, where +1 denotes clockwise. Using this δ and equation (6) we can obtain the estimate of the $\Delta\theta$ as the following:

$$\Delta\hat{\theta} = \delta \, | \, \hat{E}(\Delta\theta) \, | \, . \tag{7}$$

3.3. Scale Adaptation

Scale adaptation is a different issue from the orientation estimation. For scale adaptation, most of the object is not likely to be inside the tracking window since the object's size changes. Because of this, the assumption in Section 3.2 is not possible. Therefore, we propose a new type of descriptor for scale and adapt to scale using this descriptor.

Let σ be the relative distance from the center ("distance from the center"/"distance from the center to the boundary"), then we divide the target candidate window into areas according to σ , i.e. scale divisions. Since in mean shift tracking the mean of weights is the new position for each iteration, the weights value is the contribution of the pixel to the new position, and therefore the average of the weights for each scale divisions are equal to the amounts those scale divisions contributed to the new position of the iteration. Using this approach, we can define the following new descriptor for scale:

$$\tilde{\sigma} = \sum_j w_{\mathrm{avg},j} \sigma_{\mathrm{mean},j} \, , \tag{8}$$

where $w_{\mathrm{avg},j}$ and $\sigma_{\mathrm{mean},j}$ are the average of the weight defined in equation (1) and mean of the σ s of each pixel inside the j^{th} scale division, respectively. This descriptor shows where the mean of the contribution respect to the scale is located. Using this descriptor, we adapt the current target candidate to match $\tilde{\sigma}_{\mathrm{candidate}}$ (the current $\tilde{\sigma}$) to be $\tilde{\sigma}_0$ (the initial $\tilde{\sigma}$ of the model) as the following:

$$\sigma_{\mathrm{new}} = \frac{\tilde{\sigma}_{\mathrm{candidate}}}{\tilde{\sigma}_0} \sigma_{\mathrm{old}} \, . \tag{9}$$

By matching the current and initial $\tilde{\sigma}$ s, we can keep track of the target object's scale change

3.4. Summary

When tracking objects, mean shift finds the most probable position of the target object through iteration. During this iteration, the pixel information inside the tracking window is not reliable. This is especially true when the iteration is moving the target candidate for significant amounts. Thus, our method estimates orientation and adapts to scale only when the target candidate is moving slowly, i.e. only when the iteration is almost finished.

Given the object model $\mathbf{q}$ (the kernel, histogram and the angular histograms) and $\tilde{\sigma}_0$, the tracking algorithm is as follows:

1. Create the target candidate model $\mathbf{p}$
2. Compute the $\Delta\mathbf{x}$ using $\mathbf{q}$ (equation (2))
3. $\mathbf{x}_{\mathrm{new}} \leftarrow \mathbf{x}_{\mathrm{old}} + \Delta\mathbf{x}$
4. If $\Delta\mathbf{x} > \varepsilon'$ go to 1.
5. $\sigma_{\mathrm{new}} \leftarrow \dfrac{\tilde{\sigma}_{\mathrm{candidate}}}{\tilde{\sigma}_0} \sigma_{\mathrm{old}}$ (equation (8))
6. $\theta_{\mathrm{new}} \leftarrow \theta_{\mathrm{old}} + \Delta\hat{\theta}$ (equation (7))
7. Repeat steps 1 to 6 until $\Delta\mathbf{x} < \varepsilon''$,

where ε' is the threshold for performing orientation estimation and scale estimation and ε'' is the threshold for convergence of the mean shift.

4. Experiments

To demonstrate the performance of the proposed method, we tested with two image sequences. The proposed method was implemented in C++ using $16 \times 16 \times 16$ RGB histogram, 16 orientation divisions, 10 scale divisions, 2 for ε' and 0.9 for ε'' and was tested on a dual core 2.0GHz PC.

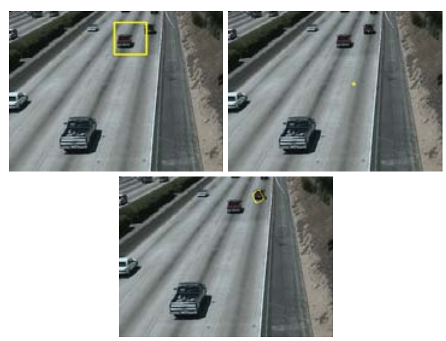
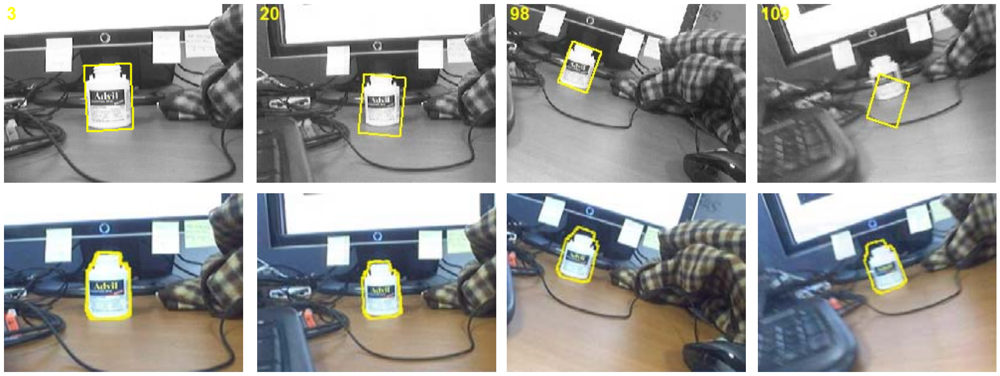


Figure 2. Result of the traditional mean shift tracker (top-left), traditional mean shift tracker with the 10% scale adaptation (top-right) and the proposed method (bottom).

Figure 2 is a single frame of an image sequence of cars moving on some highway. Using this image sequence, the proposed algorithm was compared with the traditional mean shift with and without 10% scale adaptation. The result of the original mean shift algorithm without scale adaptation on the top-left resulted



#3 #20 #98 #109

Figure 3. Tracking results for the traditional particle filter (above) and the proposed method (below).

in tracking failure since it failed to adapt to scale change and another similar object entered the target candidate region. In the top-right, the 10% method failed to adapt to scale change and shrank to a small box. Our method on the bottom shows some minor errors in tracking the orientation of the object due to the drastic change in scale and some change in the viewpoint, but still succeeded in following the target object.

Figure 3 is the tracking result of the proposed method compared with a particle filter tracker on an image sequence of a medicine bottle captured using a handheld webcam. The image sequence is very shaky and therefore the medicine bottle shows fast random movements. The particle filter implementation method was implemented as a traditional particle filter with grayscale value as the pixel information. As shown in frame 20 in both tracking results, the particle filter fails to adapt to fast scale change and results in some inaccurate result, whereas the proposed method succeeds. In frame 109, the webcam was moved very fast and we can see that the proposed method still tracks the object very neatly, whereas the traditional particle filter fails to do so. Moreover, the particle filter was tested with 200 particles and was much slow compared to the proposed method.

5. Conclusion and Future Work

The method proposed in this paper allows the traditional mean shift algorithm to track objects which vary in scale and orientation and be more robust to background clutters. This is made possible by the use of the kernel constructed based on object region masks, orientation estimation through the reconstruction of the orientation probability distribution based on the object models histogram for each orientation range, and the newly proposed descriptor for scale which allows scale adaptation. Based on test results, the method showed good results.

However, the method has its problems on some unstable estimation results on orientation. We believe this problem can be solved using some filtering techniques on the estimation results. We are currently working to improve this new method to be more robust and to apply learning techniques so that this method would be able to track viewpoint changes as well.

6. Acknowledgement

This work has been supported by the Korean Ministry of Knowledge Economy, Samsung Techwin, and BK 21 program.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:564-575, 2003.

[2] R. Collins, Mean-shift blob tracking through scale space, in *IEEE Conf. on Computer Vision and Pattern Recognition,* 2003.

[3] M. Isard and A. Blake, CONDENSATION – Conditional Density Propagation for Visual Tracking, *International Journal on Computer Vision, pp. 5-28, Vol. 1(29)*, 1998

[4] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, Color-Based Probabilistic Tracking, *in Proc. European Conf. on Computer Vision, Copenhagen, Denmark, volume I, 2002, pp. 661–675.*

[5] L. Jongwoo, D. Ross, L. Ruei-Sung, Y. Ming-Hsuan, Incremental Learning for Visual Tracking, *Advances in Neural Information Processing Systems*, 2004.

[6] A. Yilmaz, Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection, in *IEEE Conf. on Computer Vision and Pattern Recognition, 2007*