# Modified Mean Shift Tracking with Variational Normalization Coefficient

Shaozhuo Zhai†, Yuehu Liu†, Xinzhao Li†, Zhichao Cui†

†Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
No.28, Xianning West Road, Xi'an, Shaanxi, 710049, P.R. China
Email: shaozhuozhai@stu.xjtu.edu.cn, liuyh@mail.xjtu.edu.cn

**Abstract**– Mean shift algorithm has a promising performance on object tracking due to its simplicity and robustness, while the compromise between standard kernel density estimation and visual target representation degrades the tracking performance. By using anisotropic kernels defined on object shape and introducing kernel orientation in KDE to describe object rotation, we propose a target representation with better precision. Then based on this representation, a modified mean shift iteration procedure is derived by minimizing the distance function with variational normalization coefficient. Experiments demonstrate that the presented algorithm tracks target with more accuracy without increasing the computational complexity compared with classical mean shift tracking algorithm.

## 1. Introduction

Object tracking is an important task in computer vision. Among various tracking methods, kernel-based object tracking [1] is well known for its low computational complexity and robustness to background clutter, partial occlusion, motion blur and local deformation. It uses kernel density estimation (KDE) to describe the target, and minimize the distance between target model and its candidates to locate the target based on mean shift algorithm, which was originally developed for data clustering [6] and then successfully applied to image segmentation [3] and object tracking [1].

Despite its promising performance on object tracking, there are some limitations of the traditional mean shift tracking method. First of all, the object kernel is defined on a geometric shape, usually a rectangle or an ellipse as figure 1-(a) and 1-(b) shows, which contains non-object regions. This biases the motion estimation and results in loss of the tracked object. Yilmaz [2] uses the level set function as an asymmetric object kernel to conquer this limitation. Secondly, the object scale and orientation change is not well handled. Most modifications methods of the mean shift tracking [4, 5] are focused on this defect.

In this paper, we address the problem by presenting a more accurate target representation which uses object shape kernel and introduces kernel orientation in KDE, and deriving a modified mean shift tracking algorithm. Section 2 describes the novel target representation in detail. Section 3 derives the iterative equation of mean shift procedure by minimizing a distance function with variational normalization coefficient, and proposes a complete tracking algorithm. Experimental results of the proposed method are given in section 4. Finally, the paper concludes in section 5.

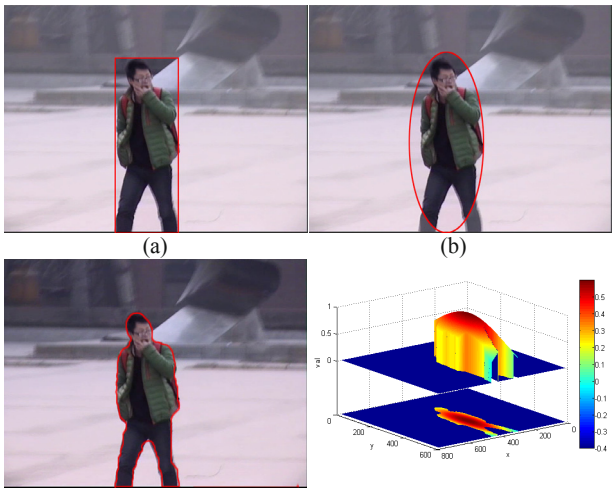

Figure1. (a) Object kernel defined on a rectangular region. (b) Object kernel defined on an elliptic region. (c) Object kernel defined on its shape, which is an irregular region. (d) The value of object shape kernel given in (c) at each pixel generated from Gaussian kernel function

## 2. Target Representation Based on KDE

The foundation of target representation for mean shift tracking is KDE also termed the Parzen–Rosenblatt window method, which is a non-parametric way to estimate the probability density function of a random variable. Given a set of sample points $\{\mathbf{x}_i \mid i=1,\ldots,n\}$ in $d-$ dimensional space, the kernel density estimator for $\mathbf{x}$ with kernel function $k(\cdot)$ and bandwidth matrix $\mathbf{H}$ are given by

$$\hat{f}_{\mathbf{H},K}(\mathbf{x}) = \frac{c_{k,d}}{|\mathbf{H}|^{1/2}\sum_{i=1}^{n} w(\mathbf{x}_i)} \sum_{i=1}^{n} k\left(d(\mathbf{x},\mathbf{H},\mathbf{x}_i)\right) w(\mathbf{x}_i) \qquad (1)$$

where $d(\mathbf{x},\mathbf{H},\mathbf{x}_i) = (\mathbf{x}-\mathbf{x}_i)^T \mathbf{H}^{-1}(\mathbf{x}-\mathbf{x}_i)$ , $\mathbf{H} = diag[h_1^2,\ldots,h_d^2]$ , $c_{k,d}$ is the normalization coefficient of kernel function $k(\cdot)$ and $w(\mathbf{x}_i)$ is the weight of sample point $\mathbf{x}_i$ .

Traditional mean shift tracking method uses probability density function (pdf) in feature space to represent the target. Both the pdfs of target model $\mathbf{q}$ and target

candidate $\mathbf{p}(\mathbf{y})$ defined at location $\mathbf{y}$ are estimated from image data using KDE and discretely represented with $m-$bin histogram as follows

$$\hat{\mathbf{q}} = \{\hat{q}_u \mid u = 1,\ldots,m; \quad \sum_{u=1}^{m} \hat{q}_u = 1\} \tag{2}$$

$$\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y}) \mid u = 1,\ldots,m; \quad \sum_{u=1}^{m} \hat{p}_u(\mathbf{y}) = 1\} \tag{3}$$

where $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}(\mathbf{y})$ are the KDE of $\mathbf{q}$ and $\mathbf{p}(\mathbf{y})$ respectively. $\hat{q}_u$ and $\hat{p}_u(\mathbf{y})$ are computed by

$$\hat{q}_u = C\sum_{i=1}^{n} k\left(\left\|\mathbf{x}_i^*\right\|^2\right)\delta\left[b(\mathbf{x}_i^*) - u\right] \tag{4}$$

$$\hat{p}_u(\mathbf{y}) = C_{\mathbf{H}}\sum_{i=1}^{n_{\mathbf{H}}} k\left(d(\mathbf{y},\mathbf{H},\mathbf{x}_i)\right)\delta\left[b(\mathbf{x}_i) - u\right] \tag{5}$$

where $d(\mathbf{y},\mathbf{H},\mathbf{x}_i) = (\mathbf{y}-\mathbf{x}_i)^T\mathbf{H}^{-1}(\mathbf{y}-\mathbf{x}_i)$, $C = 1/\sum_{i=1}^{n} k\left(\left\|\mathbf{x}_i^*\right\|^2\right)$, $C_{\mathbf{H}} = 1/\sum_{i=1}^{n_{\mathbf{H}}} k\left(d(\mathbf{y},\mathbf{H},\mathbf{x}_i)\right)$, and $b(\mathbf{x}_i) = u_i$ is the bin index in the quantized feature space for pixel at location $\mathbf{x}_i$.

### 2.1. Object Shape Kernel

Equation 1 and 5 indicate that the target is estimated with the kernel $k_j(\cdot)$ in 3D joint space generated from the product of kernel $k_s(\cdot)$ in 2D spatial space and kernel $k_f(\cdot)$ in 1D quantized feature space, which is

$$k_j(\mathbf{x},u) = k_s(\mathbf{x})k_f(u) \tag{6}$$

where delta function is used as $k_f(\cdot)$ for simplicity. However, only the pdf $\{\hat{p}_u\}$ in 1D quantized feature space is used to represent the target instead of the pdf $\{\hat{p}_{\mathbf{x},u}\}$ in 3D joint space. In another word, the target is represented by just one specific point $\mathbf{y}$ rather than all points that belong to it. Furthermore, kernel $k_s(\cdot)$ is defined on a geometry shape related with location $\mathbf{y}$, which is usually a rectangle or ellipse centered at $\mathbf{y}$ as shown in figure 1-(a) and 1-(b). All these compromises mentioned above between visual object representation and standard KDE result in a rough description of target and degrade the tracking performance.

Unlike the classical mean shift tracking method, we define the kernel on object shape as figure 1-(c) and 1-(d) shows to obtain a more precise target representation. In this situation, the calculation of location $\mathbf{y}$ that represents the target for irregular shape becomes a new problem which is solved in section 3.

### 2.2. Kernel Orientation

In equation 3, parameter $\mathbf{y}$ and $\mathbf{H}$ describes the location and scale change of target. And the target representation is rotational invariant with an isotropic kernel. As for object shape kernels which are anisotropic most of the time, it is necessary to introduce the kernel orientation angle denoted by $\theta$ to describe the object rotation. Then, the pdf of target candidate is estimated as

$$\hat{p}_u(\mathbf{y},\theta) = C_{\mathbf{H}}\sum_{i=1}^{n_{\mathbf{H}}} k\left(d(\mathbf{y},\theta,\mathbf{H},\mathbf{x}_i)\right)\delta\left[b(\mathbf{x}_i) - u\right] \tag{7}$$

where $d(\mathbf{y},\theta,\mathbf{H},\mathbf{x}_i) = (\mathbf{y}-\mathbf{x}_i)^T\mathbf{R}^T\mathbf{H}^{-1}\mathbf{R}(\mathbf{y}-\mathbf{x}_i)$, and $C_{\mathbf{H}} = 1/\sum_{i=1}^{n_{\mathbf{H}}} k\left(d(\mathbf{y},\theta,\mathbf{H},\mathbf{x}_i)\right)$, $\mathbf{R} = \begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix}$.

## 3. Target Localization

Define the distance between two discrete distributions based on Bhattacharyya coefficient, then the object tracking problem becomes an optimization problem as bellow

$$\min_{\mathbf{y},\theta} d\{\hat{\mathbf{p}}(\mathbf{y},\theta),\hat{\mathbf{q}}\} \tag{8}$$

where $d\{\hat{\mathbf{p}}(\mathbf{y},\theta),\hat{\mathbf{q}}\} = \sqrt{1-\rho\{\hat{\mathbf{p}}(\mathbf{y},\theta),\hat{\mathbf{q}}\}}$ is the sample estimate of the distance between $\mathbf{p}$ and $\mathbf{q}$, and $\rho\{\hat{\mathbf{p}}(\mathbf{y},\theta),\hat{\mathbf{q}}\} = \sum_{u=1}^{m}\sqrt{\hat{p}_u(\mathbf{y},\theta)\hat{q}_u}$ is the sample estimate of the Bhattacharyya coefficient between $\mathbf{p}$ and $\mathbf{q}$.

The above optimization is also equivalent to

$$\max_{\mathbf{y},\theta} \rho\{\hat{\mathbf{p}}(\mathbf{y},\theta),\hat{\mathbf{q}}\} \tag{9}$$

### 3.1. Modified Mean Shift Iteration

Due to the compromises mentioned in section 2.1, there is a significant difference between the normalization coefficient $C_{\mathbf{H}}$ in equation 7 and $c_{k,d}$ in equation 1, which is $C_{\mathbf{H}}$ varies with object shape and scale while $c_{k,d}$ is a constant. Consider $C_{\mathbf{H}}$ as a variable instead of a constant, and let the partial derivative be zero, which is

$$\frac{\partial}{\partial \mathbf{y}}\rho\{\hat{\mathbf{p}}(\mathbf{y},\theta),\hat{\mathbf{q}}\} = \mathbf{0} \tag{10}$$

$$\frac{\partial}{\partial \theta}\rho\{\hat{\mathbf{p}}(\mathbf{y},\theta),\hat{\mathbf{q}}\} = 0 \tag{11}$$

Then we have

$$\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y},\theta,\mathbf{H},\mathbf{x}_i)\right)(\mathbf{x}_i - \mathbf{y})w_i = \mathbf{0} \tag{12}$$

$$(h_1 - h_2)\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y},\theta,\mathbf{H},\mathbf{x}_i)\right)(A_i\tan 2\theta - 2B_i)w_i = 0 \tag{13}$$

where $A_i = (x_{i,1}-y_1)^2 - (x_{i,2}-y_2)^2$, $B_i = (x_{i,1}-y_1)(x_{i,2}-y_2)$, and $w_i = w_i^{(1)} - w^{(0)}$, $w^{(0)} = \rho\{\hat{\mathbf{p}}(\mathbf{y},\theta),\hat{\mathbf{q}}\}$,

$$w_i^{(1)} = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{y},\theta)}}\delta\left[b(\mathbf{x}_i) - u\right] \tag{14}$$

According to the classical mean shift tracking algorithm, the iterative equation for location $\mathbf{y}$ should be

$$\mathbf{y}^{(k+1)} = \frac{\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) \mathbf{x}_i w_i}{\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) w_i} \tag{15}$$

However, the sample weight $w_i$ changes from $w_i^{(1)}$ to $w_i^{(1)} - w^{(0)}$ compared with classical mean shift tracking method, which is no longer nonnegative for all $i$. Therefore, the convergence of mean shift iteration based on equation 15 is not guaranteed [3, 6].

The deduction process indicates that the weight change item $-w^{(0)}$ results from the variation of $C_{\mathbf{H}}$ which is related to the object shape. In another word, the weight change item $-w^{(0)}$ characterizes the influence of object shape variation on location $\mathbf{y}$ and orientation $\theta$. Thus, the original iteration procedure can be split into two parts. Part 1 is to obtain location $\mathbf{y}$ corresponding to current object shape by iterating with equation 16 until convergence. And part 2 is locate the target in spatial space by iteration equation 17 till convergence. In this way, the convergency of both equation 16 and 17 is guaranteed and equation 15 as well as equation 10 holds after their convergence.

$$\mathbf{y}^{(k+1)} = \frac{\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) \mathbf{x}_i}{\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right)} \tag{16}$$

$$\mathbf{y}^{(k+1)} = \frac{\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) \mathbf{x}_i w_i^{(1)}}{\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) w_i^{(1)}} \tag{17}$$

As for angle $\theta$, if $h_1 \neq h_2$, $\theta$ can be any value, and if $h_1 \neq h_2$, we obtain

$$\tan 2\theta^{(k+1)} = \frac{2\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) B_i w_i}{\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) A_i w_i} \tag{18}$$

The iteration procedure for $\theta$ splits in the same way as equation 15, which is

$$\tan 2\theta^{(k+1)} = \frac{2\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) B_i}{\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) A_i} \tag{19}$$

$$\tan 2\theta^{(k+1)} = \frac{2\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) B_i w_i^{(1)}}{\sum_{i=1}^{n_{\mathbf{H}}} g\left(d(\mathbf{y}^{(k)}, \theta^{(k)}, \mathbf{H}, \mathbf{x}_i)\right) A_i w_i^{(1)}} \tag{20}$$

### 3.2. The Complete Tracking Algorithm

In summary, the complete modified mean shift tracking algorithm is presented below.

Given:

The image that contains the target and its corresponding shape mask $\Re$.

Initialization to generate the target model:

1. Initialize location $\mathbf{y}_0^{(0)}$ and orientation $\theta_0^{(0)}$ for the target model with the center and orientation of the minimum enclosing rectangle.
2. Compute location $\mathbf{y}_0^{(j+1)}$ and orientation $\theta_0^{(j+1)}$ of target model by equation 16 and 19.
3. If $\left\|\mathbf{y}_0^{(j+1)} - \mathbf{y}_0^{(j)}\right\| < \varepsilon_{\mathbf{y}}$ and $\left\|\theta_0^{(j+1)} - \theta_0^{(j)}\right\| < \varepsilon_{\theta}$, evaluate $\left\{\hat{q}_u(\mathbf{y}_0, \theta_0) | u = 1, \ldots, m\right\}$ according to equation 7, generate a normalized shape mask $\Re_0$ with location $\mathbf{y}_0 = \mathbf{0}$ and orientation $\theta_0 = 0$ by rotating and translating the given shape mask $\Re$, and then go to step 4. Otherwise, $j \leftarrow j+1$ and go to step 2.

Tracking target frame by frame:

4. Initialize the object location $\mathbf{y}^{(0)}$ and orientation $\theta^{(0)}$ in the current frame with results from the last frame.
5. Calculate $\left\{\hat{p}_u(\mathbf{y}^{(k)}, \theta^{(k)}) | u = 1, \ldots, m\right\}$ with equation 7.
6. Derive the pixel weights $\left\{w_i^{(1)} \mid i = 1, \ldots, n_{\mathbf{H}}\right\}$ with equation 14.
7. Compute object location $\mathbf{y}^{(k+1)}$ and orientation $\theta^{(k+1)}$ for target candidates by equation 17 and 20.
8. If $\left\|\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}\right\| < \varepsilon_{\mathbf{y}}$ and $\left\|\theta^{(k+1)} - \theta^{(k)}\right\| < \varepsilon_{\theta}$, stop. Otherwise, $k \leftarrow k+1$ and go to step 5.

In step 5, 6 and 7, the sample dataset for computing is $\{\mathbf{x}_i \mid \mathbf{R}(\theta^{(k)})(\mathbf{x}_i - \mathbf{y}^{(k)}) \in \Re_0\}$ despite scale change. And in step 2 and 7, motion continuity is used to obtain angle $\theta$. The proposed algorithm can be extended for adaptive shape change by adding an optional step of re-initialization as step 2 and 3 after step 8.

## 4. Experimental Results

RGB color space is used as the feature space and quantized into 16×16×16 bins in our experiments. The proposed algorithm is experimented on both synthetic video sequences and real video sequences compared with classical mean shift tracking algorithm. Figure 2 gives the tracking results of our algorithm, which demonstrates the effectiveness and efficiency of the proposed algorithm.

According to experimental result in figure 3, the iteration number of our algorithm for each frame on average is 10.04, which is slightly higher than 9.12 of classical mean shift tracking algorithm. Besides, both algorithms share the same level of time complexity for each iteration. Therefore, our algorithm preserves the time efficiency of classical mean shift tracking algorithm with better tracking accuracy and robustness.

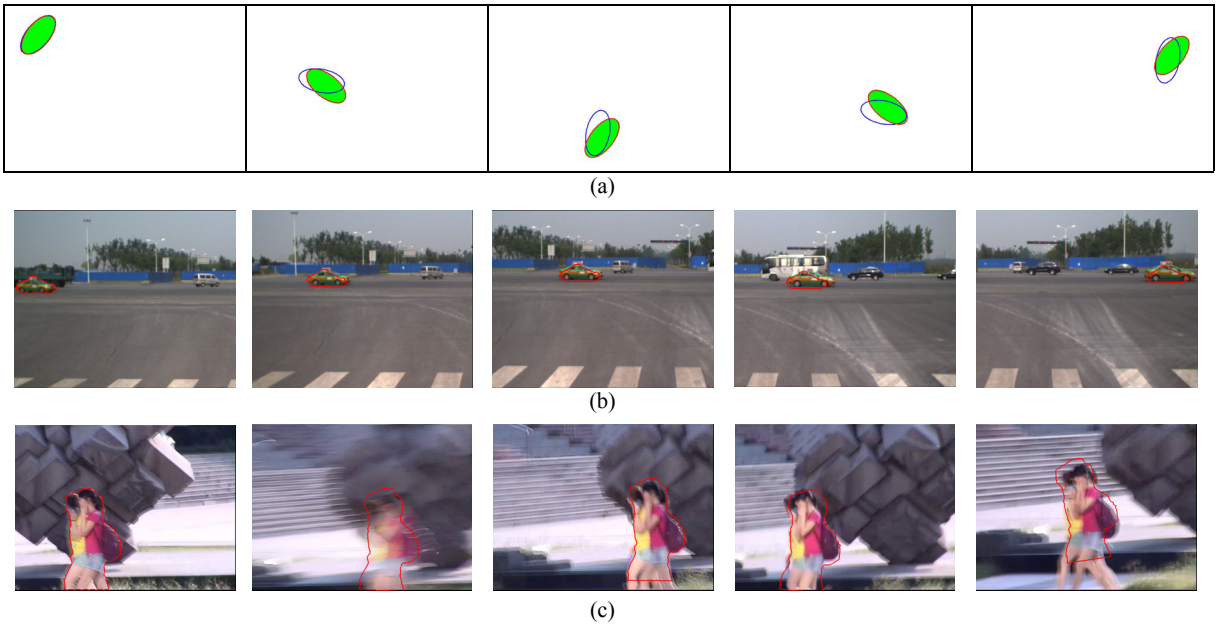

Figure 2. (a) Tracking results of the synthetic ellipse sequence. (b) Tracking results of the traffic sequence. (c) Tracking results of the pedestrian sequence. The red line in (a), (b), (c) represents the object shape kernel after convergence while the blue line in (a) represents the object shape kernel of last frame. The frames 1, 10, 20, 30, 40 are displayed.

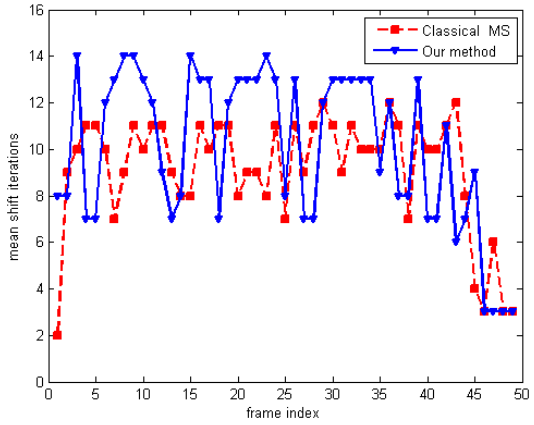


Figure 3. The iteration number for each frame of the synthetic ellipse sequence. Red line represents the classical mean shift tracking algorithm and blue line is our proposed algorithm.

## 5. Conclusions

This paper presents a more accuracy target representation by applying anisotropic kernels defined on object shape and introducing kernel orientation in KDE to describe object rotation. Based on this representation, a modified mean shift tracking algorithm is derived by minimizing the distance function with variational normalization coefficient. Experiments demonstrate that the presented algorithm tracks target with more accuracy without increasing the computational complexity compared with classical mean shift tracking algorithm. Another advantage of our proposed algorithm is that it can be extended to adapt shape change during tracking, which would be our future research.


## Acknowledgments

The authors would like to thank NSFC for support under the grant number 91120009 , 61328303 and 61305051.


## References


[1] Dorin Comaniciu, Visvanathan Ramesh and Peter Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, no.5, pp.564-577, 2003.

[2] Alper Yilmaz, "Kernel-based object tracking using asymmetric kernels with adaptive scale and orientation selection," *Machine Vision and Applications*, vol.22, no.2, pp.255-268, 2009.

[3] Dorin Comaniciu and Peter Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.5, pp.603-619, 2002.

[4] Zoran Zivkovic and Ben Krose, "An EM-like algorithm for color-histogram-based object tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, vol.1, pp.798-803, 2004.

[5] Robert T. Collins, "Mean-shift blob tracking through scale space," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.2, pp.II-234-40, 2003.

[6] K. Fukunaga and L. Hostetler "The estimation of the gradient of a density function, with applications in pattern recognition," *IET Computer Vision*, vol.21, no.1, pp.32-40, 1975.