Adaptive Bandwidth Mean Shift Object Detection

Xiaopeng Chen
Institute of Automation, Chinese Academy of Sciences
Beijing, China
xpchen@hitic.ia.ac.cn

Haiyan Huang
Faculty of Information Science and Technology, Jiujiang University
Jiujiang, China
goodhelen406@163.com

Haibo Zheng
Institute of Automation, Chinese Academy of Sciences
Beijing, China
haibo.zheng@ia.ac.cn

Chengrong Li
Institute of Automation, Chinese Academy of Sciences
Beijing, China
lich@hitic.ia.ac.cn

Abstract— **In this paper, a novel adaptive bandwidth mean shift algorithm toward 2D object detection (ABMSOD) is proposed. It can not only identify whether an object of certain classes exists or not, but also get the scale and orientation besides position very fast. The feature histogram weighted by a kernel with adaptive bandwidth is used for representing the target object model and the candidate object model. Features such as color, texture, gradient and so on can be used. A single piece of image is enough to build a model by calculating the weighted feature histogram of the object in the image. There is no exhaustive training. The similarity of the target model and the candidate model is measured by the Bhattacharyya coefficient. After gathering the models of targets, the algorithm can be used for object detection. In the first step, the algorithm searches the whole image to find the rough positions of possible candidate objects. If the similarities are all below a certain threshold, it reports no object existence. If the similarities are above the threshold, the second step or the adaptive bandwidth mean shift search step is executed to find the best position, orientation and scale of these objects. Experiments show that it successfully detects the position, scale and orientation of objects.**

***Keywords* - object detection, ABMSOD, adaptive bandwidth mean shift, feature histogram, vision navigation**

I. INTRODUCTION

Object detection is the process of finding whether there are instances in a given image or image sequence which belong to certain object class, and if so, return the number of objects and their locations in the given image or image sequence [1]. Many factors such as intra class variation, changes of viewpoint, illumination variation, occlusion, cluttered background and so on impact the detection result greatly [2]. Despite the great success made by Paul Viola [14], generic object detection is still one of the most difficult task in computer vision. The choice and representation of common features for each object class is the main difficulty. Features which are good for a certain object class detection may be inefficient for another object class. For example, gradient and shape and contour features are only suitable for rigid object detection.

There are many existing algorithms for object detection. They vary from different features to different methods for classification. Features include color [3], texture [4], shape [5], appearance [6], edge or gradient [7], contour [8], wavelet feature [9], joint feature [10], local feature [11], scale invariant feature [12], context feature [13] and other extracted features. Popular methods include boost learning [14], support vector machine learning [15], neural network learning [16], genetic learning [17], Bayesian learning [18], Component Analysis [19], template matching [20], gabor filter feature extracting method [21], wavelet feature extracting method [9] and so on. Some of them detect only moving objects in image sequences [22], and some others detect objects from static images[14]. Many of them are task specific, and the rest are generic.

Most of them use learning, that is to say, the detection systems are trained by large exemplars. Machine learning approaches, while powerful, need a large number of exemplars under various conditions and may not work well for non-rigid objects [23]. What's more, exemplars are not easy to get. And the speed is still a bottleneck.

In this paper, we propose a novel object detection algorithm under the adaptive bandwidth mean shift framework which is our previous work proposed in [25]. It is very effective and efficient, and it has a good balance between speed and precision. It is quite suitable for object detection in vision navigation. We call it adaptive bandwidth mean shift object detection or ABMSOD for short. The ABMSOD algorithm is not a task specific but a generic detection algorithm. It can not only be used in mobile robot navigation, but also be used in other applications. It can detect objects in static images and image sequences. We use kernel weighted feature histograms to descript models and candidate models. Features can be color, texture, gradient magnitude, or even haar-like feature, etc. The Bhattacharyya coefficient is used to measure the similarity of the model and candidate model. There are mainly two steps in the algorithm: in the first step, the algorithm search roughly to get the possible positions of given object model. If the similarities of the possible target model and the given sample are above the threshold, a second step is executed to find the best location, orientation and scale that best maximize the similarity. The second step can be easily integrated with object tracking framework whenever needed. It is verified by experiments.

This paper is organized like this: section II briefly introduces the adaptive bandwidth mean shift framework. (Details of the framework can be seen in [25]), Section III derives a general object detection algorithm under adaptive bandwidth mean shift framework. Section IV gives many experiment results. We draws conclusion in Section V.

II. ADAPTIVE BANDWIDTH MEAN SHIFT

A general kernel for multivariate kernel density estimator is
$$K_{\boldsymbol{H}}(\boldsymbol{x}) = |\boldsymbol{H}|^{-\frac{1}{2}} K(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}^{-1}\boldsymbol{x}) \tag{1}$$
Where $\boldsymbol{H}$ is a symmetric matrix. Profile $K(x):[0,\infty]\to R$ satisfies these conditions: nonnegative, nonincreasing and piecewise continuous [24].

The weighted density estimate using d-variate general kernel is defined like this:
$$q(\boldsymbol{x}) = \sum_{s\in S} K_{\boldsymbol{H}}(\boldsymbol{x}-\boldsymbol{s})w(\boldsymbol{s}) \tag{2}$$
Where $S \subset X$ is a finite set that represents the search space, $w: S \to (0,\infty)$ is a weight function. The sample mean shift kernel is defined:
$$G_{\boldsymbol{H}}(\boldsymbol{x}) = |\boldsymbol{H}|^{-\frac{1}{2}} G(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}^{-1}\boldsymbol{x}) \tag{3}$$
where
$$G(x) = -K'(x) \tag{4}$$
The mean shift vector is defined:
$$\boldsymbol{m}(\boldsymbol{x}) - \boldsymbol{x} = \frac{\sum_{s\in S} G_{\boldsymbol{H}}(\boldsymbol{x}-\boldsymbol{s})w(\boldsymbol{s})\boldsymbol{s}}{\sum_{s\in S} G_{\boldsymbol{H}}(\boldsymbol{x}-\boldsymbol{s})w(\boldsymbol{s})} - \boldsymbol{x} \tag{5}$$
Paper [25] proved that when the symmetric bandwidth matrix $\boldsymbol{H}$ is definite positive, the inner product between the mean shift vector and the gradient of $q(\boldsymbol{x})$ is absolutely positive, which means that the mean shift vector points to the direction for local maximum.

If the first derivative of (2) on $\boldsymbol{H}$ or $\boldsymbol{H}^{-1}$ is zero and the second derivative is less than zero, the solution of $\boldsymbol{H}$ or $\boldsymbol{H}^{-1}$ in (6) will maximize the value of $q(\boldsymbol{x})$.
$$\begin{cases} \dfrac{\partial q(\boldsymbol{x})}{\partial \boldsymbol{H}} = 0 \\ \dfrac{\partial^2 q(\boldsymbol{x})}{\partial \boldsymbol{H}^2} < 0 \end{cases} \text{or} \begin{cases} \dfrac{\partial q(\boldsymbol{x})}{\partial (\boldsymbol{H}^{-1})} = 0 \\ \dfrac{\partial^2 q(\boldsymbol{x})}{\partial (\boldsymbol{H}^{-1})^2} < 0 \end{cases} \tag{6}$$
$S = \left\{\boldsymbol{s} \middle| (\boldsymbol{x}-\boldsymbol{s})^{\mathrm{T}}\boldsymbol{H}^{-1}(\boldsymbol{x}-\boldsymbol{s}) < r^d\right\}$ defines a super elliptic ball in d-variate space centered at point $\boldsymbol{x}$ where to search the samples.

When it is in 2D case, as $\boldsymbol{H}$ is symmetric definite positive, it can be rewritten as
$$\boldsymbol{H} = \boldsymbol{R}(\phi)\boldsymbol{Diag}(\hat{a},\hat{b})\boldsymbol{Diag}(\hat{a},\hat{b})\boldsymbol{R}^{\mathrm{T}}(\phi) \tag{7}$$
And
$$S_{2D} = \left\{\boldsymbol{s} \middle| (\boldsymbol{x}-\boldsymbol{s})^{\mathrm{T}}\boldsymbol{H}^{-1}(\boldsymbol{x}-\boldsymbol{s}) < \sigma^2\right\} \tag{8}$$
defines an ellipse centered $\boldsymbol{x}$ at whose two half axes length are $\sigma\hat{a}$ and $\sigma\hat{b}$ and whose rotate angle is ϕ. When Epanechnikov profile is used, we set σ to 1; when Gaussian profile is used, we set σ to 2.5. When σ is set, the bandwidth matrix $\boldsymbol{H}$ decides the elliptical search space. It is furthrer proved in paper [25] that the optimal bandwidth matrix for $q(\boldsymbol{x})$ is
$$\boldsymbol{H} = \frac{\lambda\sum_{s\in S} w(\boldsymbol{s})(\boldsymbol{x}-\boldsymbol{s})(\boldsymbol{x}-\boldsymbol{s})^T}{\sum_{s\in S} w(\boldsymbol{s})} \tag{9}$$
Where $\lambda_e = 4$ for Epanechnikov profile and $\lambda_g = 1$ for Gaussian profile.

Then we can draw the adaptive bandwidth mean shift algorithm for seeking the mode of (2).

1) Initialize the position $\boldsymbol{x}_0$ and bandwidth matrix $\boldsymbol{H}_0$, and then the initial search space S_0 is calculated according to (8).

2) Use (5) to shift the position $\boldsymbol{x}_0$ to $\boldsymbol{x}_1$ ($\boldsymbol{x}_1 = \boldsymbol{m}(\boldsymbol{x}_0)$), the search space would be updated as S_0' according to updated center point $\boldsymbol{x}_1$.

3) Search the maximum of $\boldsymbol{H}_1$ according to (6) or (9) in 2D case in S_0', the search space would be updated as S_1 according to updated bandwidth matrix $\boldsymbol{H}_1$.

4) If $\|\boldsymbol{x}_0 - \boldsymbol{x}_1\| < \varepsilon$ and $S_0 = S_1$, stop. Else $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_1$, $S_0 \leftarrow S_1$, $\boldsymbol{H}_0 \leftarrow \boldsymbol{H}_1$, goto 2).

III. DETECTION USING ADAPTIVE BANDWIDTH MEAN SHIFT

In order to detect an object or an object class, features that are insensitive to scale and orientation changes for representing the object model must be selected. Object model should be able to fuse features as many as possible and very easy to extend. What's more, the representation can be used further for localization of objects.

We used the adaptive bandwidth mean shift framework from our pervious work in paper [25] for object detection. Previously it was used for object tracking. However, it is also suitable for object detection from object representation to object identification and localization. It is called adaptive bandwidth mean shift object detection (ABMSOD). The details of the algorithm is given below.

A. Features for Object Model Representation

Some related work called histogram matching method should be mentioned. Color histogram was first used for object detection and object recognition in 1991 [3]. Schiele et al. [26] generalized this idea to histograms of receptive fields. In [27], Linde et al. evaluated more complex descriptor combinations, forming histograms of up to 14 dimensions. Mel [28] also developed a histogram based object recognition system that uses multiple low-level attributes such as color, local shape and texture. In [10], Chang et al. show how color cooccurrence histograms can be used for object detection, performing better than regular color histograms.

However, we found that these histograms are suitable for object identification, but have a poor performance on localization. Kernel weighted feature histogram, or kernel weighted feature probability distribution, instead, can not only take the benefit of histograms' simplicity, speed and robustness, but also take the benefit of localization ability under adaptive bandwidth mean shift framework. Many features including color, gradient magnitude, texture, shape or even haar-like feature can be integrated as a feature vector.

B. Model Representation

The object model is represented by its kernel weighted probability distribution function or kernel weighted feature histogram q in the feature space, the same as the definition given in [29]. Given an image containing the object with initial region $\hat{S}_0$ (ellipse) centered at position $\hat{\boldsymbol{x}}_0$, the initial bandwidth matrix $\hat{\boldsymbol{H}}_0$ can be calculated according to (7), where $\hat{a}_0 = \frac{x\_axis\_len}{\sigma}$, $\hat{b}_0 = \frac{y\_axis\_len}{\sigma}$, $\hat{\phi}_0$ is the rotating angle of the ellipse $\hat{S}_0$ .Usually $\hat{\phi}_0 = 0$. The feature histogram is weighted according to the Mahalanobis distance from point $\boldsymbol{s}$ to the region center $\hat{\boldsymbol{x}}_0$ by matrix $\hat{\boldsymbol{H}}_0$, The function $b: R^2 \to \{1...M\}$ associates to the pixel at location $\boldsymbol{s}$ of its bin in the feature space [29]. The probability of the feature index $u = 1...M$ of the target model is then computed as

$$\hat{q}_u = C \sum_{\boldsymbol{s} \in \hat{S}_0} \left|\hat{\boldsymbol{H}}_0\right|^{-\frac{1}{2}} K((\hat{\boldsymbol{x}}_0 - \boldsymbol{s})^{\mathrm{T}} \hat{\boldsymbol{H}}_0^{-1} (\hat{\boldsymbol{x}}_0 - \boldsymbol{s})) \delta[b(\boldsymbol{s}) - u] \quad (10)$$

Where $K(\bullet)$ is an Epanechnikov profile or a Gaussian profile.

As $\left|\hat{\boldsymbol{H}}_0\right|^{-\frac{1}{2}}$ is invariant to $\boldsymbol{s}$, (10) can be rewritten as

$$\hat{q}_u = C \sum_{\boldsymbol{s} \in S_0} K((\hat{\boldsymbol{x}}_0 - \boldsymbol{s})^{\mathrm{T}} \hat{\boldsymbol{H}}_0^{-1} (\hat{\boldsymbol{x}}_0 - \boldsymbol{s})) \delta[b(\boldsymbol{s}) - u] \quad (11)$$

where $$C = \frac{1}{\sum_{\boldsymbol{s} \in \hat{S}_0} K((\hat{\boldsymbol{x}}_0 - \boldsymbol{s})^{\mathrm{T}} \hat{\boldsymbol{H}}_0^{-1} (\hat{\boldsymbol{x}}_0 - \boldsymbol{s}))} \quad (12)$$

is the normalization constant. Once we have selected the features and build the weighted histogram according to (11), the target model can be used for object identification and localization. If there are many different appearances of the same object or object class, models of all the appearances are saved. If there are many object classes or objects to be detected, a database is built. Each record in the database represents an appearance model, one model or many models belong to an object or an object class.

Given a candidate sample in position $\boldsymbol{y}$ with bandwidth matrix $\boldsymbol{H}$, the point set S is defined by (8), its probability distribution is

$$\hat{p}_u(\boldsymbol{y}) = C_{\boldsymbol{H}} \sum_{\boldsymbol{s} \in S_{\boldsymbol{H}}} |\boldsymbol{H}|^{-\frac{1}{2}} K((\boldsymbol{y} - \boldsymbol{s})^{\mathrm{T}} \boldsymbol{H}^{-1} (\boldsymbol{y} - \boldsymbol{s})) \delta[b(\boldsymbol{s}) - u] \quad (13)$$

where $$C_{\boldsymbol{H}} = \frac{1}{\sum_{\boldsymbol{s} \in S_{\boldsymbol{H}}} |\boldsymbol{H}|^{-\frac{1}{2}} K((\boldsymbol{y} - \boldsymbol{s})^{\mathrm{T}} \boldsymbol{H}^{-1} (\boldsymbol{y} - \boldsymbol{s}))} \quad (14)$$

is the normalizeation constant which is not depend on $\boldsymbol{y}$.

C. Object Identification and Localization

Object identification and localization is to find the maximum similarity between the target model and candidate object model. The similarity is measured by the Bhattacharyya coefficient

$$\rho(\hat{p}(\boldsymbol{y}), \hat{q}) = \sum_{u=1}^{M} \sqrt{\hat{p}_u(\boldsymbol{y}) \hat{q}_u} \quad (15)$$

Paper [25] proved that seeking the maximum of (15) is the same as seeking the maximum of

$$q(\boldsymbol{y}) = \sum_{\boldsymbol{s} \in S_{\boldsymbol{H}}} w(\boldsymbol{s}) |\boldsymbol{H}|^{-\frac{1}{2}} K((\boldsymbol{y} - \boldsymbol{s})^{\mathrm{T}} \boldsymbol{H}^{-1} (\boldsymbol{y} - \boldsymbol{s})) \quad (16)$$

Where $w(\boldsymbol{s}) = \sum_{u=1}^{M} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\boldsymbol{y}_0)}} \delta[b(\boldsymbol{s}) - u]$. (17)

As the form of (16) is the same as (2), the adaptive bandwidth mean shift algorithm mentioned in section II can be applied for searching the local maximum of (16).

The identification and localization for a single appearance model can be done in the following steps.

1) Given an input image, randomly scatter enough ellipse regions whose initial center points are within the image size, so that all the ellipses cover the whole image. The size and angle of these ellipses are loaded from the target models.

2) Calculate the kernel weighted feature histograms in these ellipses and calculate the corresponding similarities with target models according to (15). Regions with similarities above the threshold are stored, and the left regions are discarded. The rough positions of detected objects are obtained.

For each region with rough sizes and positions calculated in step 2), do the following steps to find the best position, scale and orientation.

3) Calculate the bandwidth matrix $\boldsymbol{H}_0$ according to (7), with center position $\boldsymbol{y}_0$ obtained in step 2), the point set S_0 is calculated according to (8).

4) Derive the weights $\{w(\boldsymbol{s})\}_{\boldsymbol{s} \in S_0}$ according to (17).

5) Find the next location $\boldsymbol{y}_1$ of the candidate model according to $\boldsymbol{m}(\boldsymbol{x})$ in (5) :

$$\boldsymbol{y}_1 = \frac{\sum_{\boldsymbol{s} \in S_0} \boldsymbol{s} w(\boldsymbol{s}) G((\boldsymbol{y}_0 - \boldsymbol{s})^{\mathrm{T}} \boldsymbol{H}_0^{-1} (\boldsymbol{y}_0 - \boldsymbol{s}))}{\sum_{\boldsymbol{s} \in S_0} w(\boldsymbol{s}) G((\boldsymbol{y}_0 - \boldsymbol{s})^{\mathrm{T}} \boldsymbol{H}_0^{-1} (\boldsymbol{y}_0 - \boldsymbol{s}))} \quad (18)$$

the factor $|\boldsymbol{H}_0|^{-\frac{1}{2}}$ is eliminated in both denominator and numerator of (18) as it does not depend on $\boldsymbol{s}$.

6) While $\rho[\hat{p}(\boldsymbol{y}_1), \hat{q}] < \rho[\hat{p}(\boldsymbol{y}_0), \hat{q}]$

do $\boldsymbol{y}_1 = \frac{\boldsymbol{y}_1 + \boldsymbol{y}_0}{2}$ recalculate $\rho[p(\boldsymbol{y}_1), \hat{q}]$

7) Move the window center to $\boldsymbol{y}_1$, update S_0 recalculate $\{w(\boldsymbol{s})\}_{\boldsymbol{s} \in S_0}$.

8) Update the bandwidth matrix $\boldsymbol{H}_1$ according to (9).

9) Recalculate point region S_1 according to (8).

10) If the points in S_1 and S_0 are the same, goto 11); Else $\boldsymbol{y}_0 \leftarrow \boldsymbol{y}_1, S_0 \leftarrow S_1, \boldsymbol{H}_0 \leftarrow \boldsymbol{H}_1$, goto 5.

11) Calculate the kernel weighted feature histograms in these ellipses and calculate the corresponding similarities according

to (15). Regions with similarities above the threshold are stored, and the left regions are discarded. The possible candidate objects and their positions, scales and orientations are calculated. If only one object is needed to detect, just choose the one that has the maximum similarity among these detected objects.

If there are many appearance models to detect, run the first two steps for each appearance to get rough positions and sizes. And then runs the remaining steps to find the optimal positions, scales and orientations.

If tracking has to be executed after detection, run steps 3~10 with slight modification of step 3 in the initial values. For tracking utilities, the initial values are the results of previous frame.

IV. EXPERIMENTS

Performance evaluation of an object detection algorithm usually includes three aspects: correct object identification rate, object localization precision and average time consumed. The ideal way is to test all the algorithms on the same test dataset using the same metrics. Paper [30] presented many metrics for measuring the precision of object detection algorithms. Some of the useful metrics are “Area Based Precision for Frame”, “Average Object Area Recall” and “Localized Object Count Recall”. They are easy to understand but are not easy for implementation. Receiver Operating Curve (ROC) [31] is used for expressing the trade-off between the correct identification rate and false identification rate in object detection algorithms. Paper [32] proposed recall- precision rate instead of ROC for expressing the trade-off for evaluating the identification performance of object detection algorithms.

Most object detection algorithms give emphasis on object identification ability, and most existing datasets are used for test correct identification rate. However, ABMSOD gives more emphasis on the localization abilities, and so comparison among algorithms is not presented in this paper. The detection results are only compared with the ground truth, which is labeled manually using the tools from http://homepages.inf.ed.ac.uk/rbf/CAVIAR/. We use datasets of our own for testing the localization abilities. The first is doorplate image dataset, and the second is a box image dataset. Our precision error measurements are derived from [25]. The errors and mean errors of position, scale and orientation are defined like this:

$$\begin{cases} e_x = x(t) - x_{gt}(t) \\ e_y = y(t) - y_{gt}(t) \\ e_o = o(t) - o_{gt}(t) \\ e_a = (a(t) - a_{gt}(t)) / a_{gt}(t) \\ e_b = (b(t) - b_{gt}(t)) / b_{gt}(t) \end{cases} \tag{19}$$

$$\begin{cases} \overline{e}_x = (\sum_t abs(x(t) - x_{gt}(t))) / image\_count \\ \overline{e}_y = (\sum_t abs(y(t) - y_{gt}(t))) / image\_count \\ \overline{e}_o = (\sum_t abs(o(t) - o_{gt}(t))) / image\_count \\ \overline{e}_a = (\sum_t abs((a(t) - a_{gt}(t)) / a_{gt}(t))) / image\_count \\ \overline{e}_b = (\sum_t abs((b(t) - b_{gt}(t)) / b_{gt}(t))) / image\_count \end{cases} \tag{20}$$

In order to verify the effectiveness and efficiency of the ABMSOD algorithm, it is implemented on on a PC which has a Intel R4 3.0GHz CPU and a memory of 512MB. The feature used now is HSV color feature of 32 * 32 * 4. The results are shown from figure 1 to 5.

Figure. 1 Doorplates detected by ABMSOD algorithm

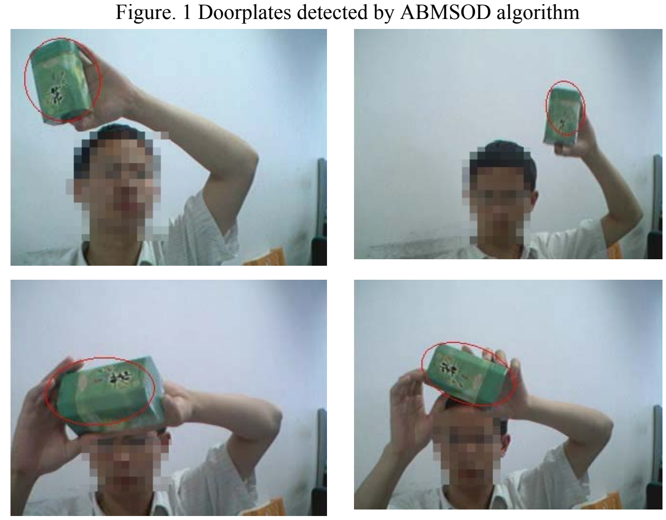

Figure. 2 Box detected in images 0, 95, 190, 285

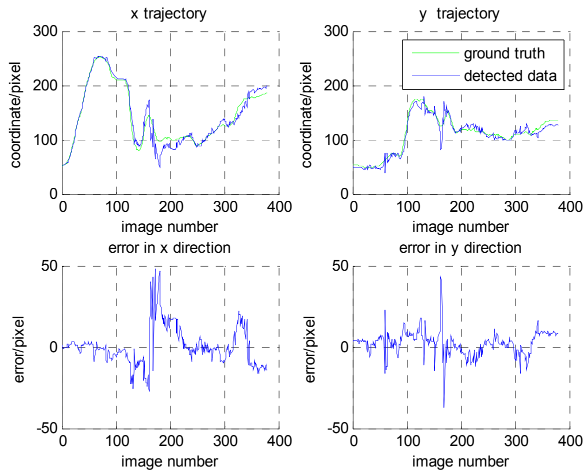

Figure. 3 Detected position and errors

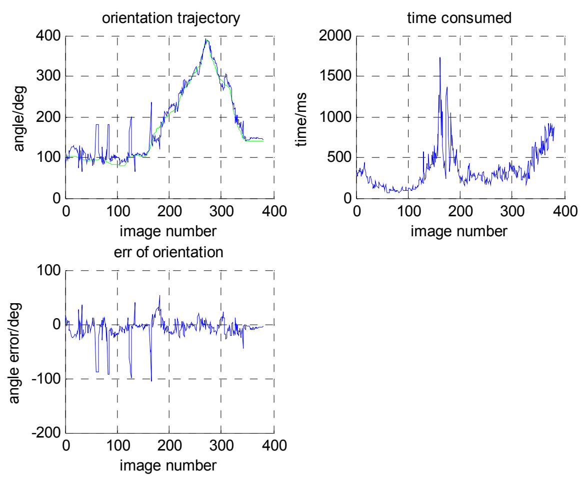

Figure. 4 Detected orientation and consumed time

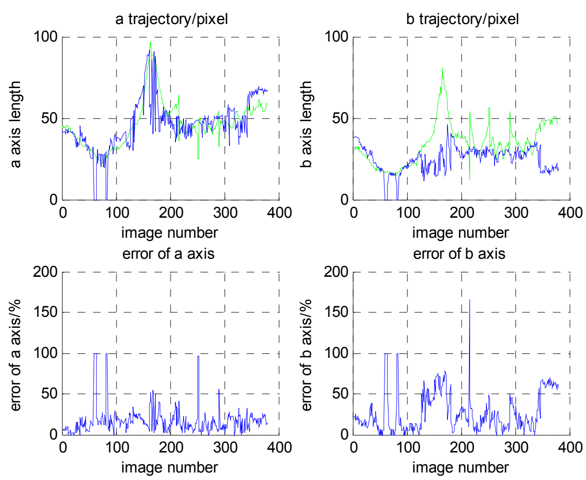

Figure. 5 Detected two axes of ellipse region and their errors

Figure 1 and 2 show the detected image results. The ellipses are located on the objects. Figure 3, 4 and 5 show the statistical detection results of the second dataset. The mean detection errors of x , y , o , a , b in the second dataset are 0.3150, 2.4016, -8.3858, 16.83%, 27.35%. The mean time consumed for each image is 345.6667ms. The blue lines are the actual detected data and the green lines are the ground truth labeled manually. In figure 3, the detected center positions are much close to its ground truth, with mean error 0.3150 and 2.4016 in pixel. In the left side of figure 4, the orientation tracking trajectory and corresponding error are drawn. The mean error is -8.3858 in degrees. The time consumed for each image is not the same. The details can be seen in the upper-right side of figure 4. The mean time is 345.6667 ms. The two axes of each ellipse describe the scale of object in two orthogonal directions. The detection results are shown in figure 5. If you look at the figure carefully, you would find that in some place both values of the two axes is zero. This is a signal of detection failure. The figures and the mean data show the detection and localization ability of ABMSOD. In all, the ABMSOD algorithm is fast and effective.

V. CONCLUSION

In this paper, we presented ABMSOD algorithm for object detection using kernel weighted feature histogram for object model representation. There are two key contributions in this paper. The first is the usage of kernel weighted feature histogram instead of histogram for object model representation. The kernel weighted feature histogram can not only take the benefit of histograms' simplicity, speed and robustness, but also take the benefit of localization ability under mean shift framework. It is easy to get the object model without exhaustive training, a single image is enough. Many features can be used without modification of the framework. The second is the ABMSOD algorithm. Besides object identification ability, the algorithm has the ability of simultaneously detecting the position, scale and orientation very fast. What's more, it can be easily used together with object tracking.

Experiments show that the algorithm is effective and efficient. ABMSOD is good for object detection and is insensitive to scale and orientation changes. It is especially suitable for landmark detection for mobile robot navigation.

Our work is at the beginning and so only color features are used in the ABMSOD algorithm. Color features may be individually different among objects of the same class. It needs improvement for generic object detection. For example, if the background has similar color distributions, the algorithm would report false positive results. The choice of similarity threshold is also vital for correct identification rate versus false identification rate. Although a single image containing target object is enough for model representation, we suggest using more images in different situations for robustness.

As the ABMSOD algorithm is suitable for fusing many features, adding new features of the same object class to the ABMSOD algorithm is our future work for improvement.

[1] Hongming Zhang, Wen gao, Xilin Chen, "Object detection using spatial histogram features", Image and Vision Computing, vol 24, 2006, pp. 327-341.

[2] Zhang huaifeng, He xiangjian and Wu Qiang. “Generic object detection: A survey,”Journal of Yunnan Nationalities University (Natural Science Edition), 2006, vol. 15, no. 4, pp. 261- 267.
[3] M. Swain and D. Ballard, “Color indexing,” IJCV7, pp. 11–32, 1991.
[4] Timo Ojala, Matti PietikaÈ inen, and Topi MaÈenpaÈaÈ. “Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns,” IEEE Trans. On Pattern Analysis and Machine Intelligence, 2002, vol. 24, no. 7, pp. 971- 987.
[5] Bastian Leibe, Ales Leonardis, and Bernt Schiele. “Combined Object Categorization and Segmentation with an Implicit Shape Model,” ECCV workshop in statistical learning and computer vision, 2004.
[6] Johan Thureson and Stefan Carlsson, “Appearance Based Qualitative Image Description for Object Class Recognition,” Proc. ECCV 2004, pp. 518- 529.
[7] Navneet Dalal and Bill Triggs, “Histograms of Oriented Gradients for Human Detection,” IEEE Proc. Of CVPR 2005.
[8] Jamie Shotton, Andrew Blake, and Roberto Cipolla, “contour based learning for object detection,” Proc. Of Tenth IEEE International Conference on Computer Vision, 2005, vol. 1, pp. 503- 510.
[9] Robin N. Strickland, and Hee Il Hahn, “Wavelet Transform Methods for Object Detection and Recovery,” IEEE Trans. On Image Processing, vol. 6, no. 5, 1997, pp. 724- 735.
[10] Peng Chang and John Krumm,“Object recognition with color cooccurrence histograms,” IEEE Conf. on CVPR, 1999.
[11] Krystian Mikolajczyk, Bastian Leibe, and Bernt Schiele, “Local Features for Object Class Recognition,” IEEE Proc. Of ICCV 2005.
[12] David G. Lowe, “Object Recognition from Local Scale-Invariant Features,” Proc. Of ICCV 1999.
[13] Lucas Paletta and Christian Greindl, “Context Based Object Detection from Video,” ICVS 2003, pp. 502- 512.
[14] Paul Viola, Michael Jones, “Rapid Object Detection Using a Boosted Cascade of Simple Features,” IEEE Computer Vision and Pattern Recognition, Vol I, 2001, pp. 511- 518.
[15] Edgar Osuna, Robert Freund and Federico Girosit, “Training Support Vector Machines: an Application to Face Detection,” IEEE proc. Of CVPR 1997, pp. 130- 136.
[16] H. Rowley, *S* Baluja, T. Kanade. Neural network-based face detection, EEE Trans. PAMI, 20(1): 23-38, 1998
[17] Daniel Howard, Simon C. Roberts, and Conor Ryan, “The Boru Data Crawler for object detection tasks in Machine Vision,” EvoWorkshops 2002, pp. 222–232.
[18] Henry Schneiderman, “Learning a Restricted Bayesian Network for Object Detection,” IEEE Proc. Of CVPR 2004,
[19] ALI S and Mubarak S, “A Supervised Learning Framework for Generic Object Detection in Images,”in Proc. Of the tenth IEEE ICCV, 2005, pp. 1347- 1354.
[20] AParna Lakshmi ratan,W.Eric L. Grimson and William M. Wells III, “Object Detection and Localization by Dynamic Template Warping,” International Journal of Computer Vision, 2000, vol. 36, no. 2, pp. 131- 147.
[21] Anil K. Jain and Nalini K. Ratha, “Object detection Using Gabor Filters,”Pattern Recognition, 1997.
[22] Michal Irani and P. Anandan, “A Unified Approach to Moving Object Detection in 2D and 3D scenes,” IEEE Trans. On PAMI, vol. 20, no. 6, 1998, pp. 577- 589.
[23] Jiebo Luo and David Crandall, “Color Object Detection Using Spatial-Color Joint Probability Functions,” IEEE Transactions on Image Processing, vol. 15, no. 6, June, 2006, pp. 1443- 1453.
[24] Cheng, Y., “Mean Shift, Mode Seeking, and Clustering,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17(8), 1995, pp. 790-799.
[25] Chen Xiaopeng, Luo Yangyu, Li Gongyan and Li Chengrong, “Adaptive Bandwidth Mean Shift Object Tracking,” P1041, RAM 2008.
[26] B. Schiele and J. L. Crowley, “Recognition without correspondence using multidimensional receptive field histograms,” International Journal of Computer Vision, vol. 36, no. 1, pp. 31–50, 2000.
[27] O. Linde and T. Lindeberg, “Object recognition using composed receptive field histograms of higher dimensionality,” in 17th International Conference on Pattern Recognition, ICPR’04, 2004.
[28] B. Mel, “SEEMORE: Combining Color, Shape and Texture Histogramming in a Neurally Inspired Approach to Visual Object Recognition,” Neural Computation, vol. 9, pp. 777–804, 1997.
[29] Comaniciu, D., Ramesh, V. and Meer, P., “Kernel- Based Object Tracking,” IEEE Transactions on Pattern And Machine Intelligence, 25(5): 564- 577, May 2003.
[30] VY Mariano, J Min, JH Park et al, “Performance evaluation of object detection algorithms,” in 16th International Conference on Pattern Recognition, 2002, pp. 965- 969.
[31] H. V. Trees, Detection, Estimation, and Modulation Theory. New York: Wiley, 2001.
[32] Shivani Agarwal and Dan Roth, “Learning a Sparse Representation for Object Detection,” ECCV 02..