The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Rapid and Precise Object Detection based on Color Histograms and Adaptive Bandwidth Mean Shift

Xiaopeng Chen, Qiang Huang, *Member, IEEE,* Peng Hu, Min Li, *Member, IEEE,* Ye Tian and Chen Li

Abstract—**Speed and precision are important for object detection algorithms. In this paper, a novel object detection algorithm based on color histogram and adaptive bandwidth mean shift is proposed. The algorithm is capable of detecting objects rapidly and precisely. It is composed of two stages: a rough detection stage and a precise detection stage. At the rough detection stage, histogram back projection and thresholding are applied to fast object identification and rough global localization. At the precise detection stage, the precise position, size and orientation are derived under the adaptive bandwidth mean shift framework. Experiments verify that the algorithm is able to detect the size, position and orientation of general objects rapidly and precisely.**

I. Introduction

OBJECT detection has been paid lots of attention in computer vision and pattern recognition communities in recent years. It also plays an important role in robotic vision. These algorithms are used for landmark detection, obstacle avoidance, localization, map building and navigation, etc. In robotic vision field, object detection algorithms should be fast enough to meet the real time requirements and keep a good balance between speed and effectiveness.

Among many object detection algorithms, one category of them is color histograms [1, 2] based object detection or/and tracking algorithms using mean shift as searching algorithm. Histograms are regional features insensitive to shapes, changes in view direction, scale and occlusion [3]. What's more, histograms based approaches are popular for their robustness, simplicity and speed. Mean shift is a real time algorithm for seeking local optimum. It was first introduced to computer vision by Comaniciu [4] and has received great success in image segmentation, object tracking, filtering and object detection, etc. However, the mean shift algorithm cannot reach global optimum[5]. What's more, neither the orientation nor the scale of objects can be obtained by mean shift algorithm.

Manuscript received December 27, 2008. This work was supported by National High-Technology Research and Development Program (No 2007AA04Z241) and the National Natural Science Foundation of China (No. 60874048)

X. Chen is with the School of Mechatronics, Beijing Institute of Technology, Beijing, 100081, China (e-mail: xpchen@bit.edu.cn)

Q. Huang is with the School of Mechatronics, Beijing Institute of Technology, Beijing, 100081, China (e-mail: qhuang@bit.edu.cn).

P. Hu is with the Hitech Innovation Center, Chinese Academy of Sciences, Beijing, 100193, China (e-mail: phu@hitic.ia.ac.cn)

M Li is with the School of Mechatronics, Beijing Institute of Technology, Beijing, 100081, China(e-mail: min@bit.edu.cn)

Y Tian and C Li are students of the School of Mechatronics, Beijing Institute of Technology, Beijing, 100081, China (e-mail: tianye7248@bit.edu.cn, ele-crazy@126.com)

Most existing histogram based algorithms calculate candidate histogram features densely on image regions to reach global optimum, such as paper [2] [6]. In order to find the positions and sizes of objects in an image, the conventional way is to compute candidate models using a sliding window on the image, each window location is treated equally. The window slides horizontally and vertically from left to right, top to bottom. This is repeated until all the image pixels are visited and all the possible scales are visited. Each time the window slides, the histogram of the sub-image within this window is calculated and is used for matching with the model histogram. This is an expensive process. Though Porikli F [6] proposed the integral histogram to speed up the calculation, the amount of computation is still prohibitive. Even if no object exists in the image, the integral histogram based search algorithm still consumes the same amount of time. In paper [1], instead of the sliding window, five randomly scattered windows are located in the image for initial localization. But the author didn't answer how to choose the sizes of the five windows.

In order to detect the scale and orientation under the mean shift framework, there many good works including scale-mean shift[7], EM-like mean shift[8], and Adaptive Bandwidth Mean Shift[9]. The ABMS is our previous work. The advantages of ABMS versus other works are illustrated in detail in [9].

In this paper, a novel object detection algorithm based on color histograms and adaptive bandwidth mean shift is proposed in this paper to reach global optimum quickly and to detect the scale and orientation of objects. There are two stages in our approach, the rough detection stage and the precision stage. The rough detection stage is a good way to reach global optimum quickly. The adaptive bandwidth mean shift is good for detecting the scale, orientation as well as position of objects quickly and precisely. At the rough detection stage, the back projection based saliency mechanism is introduced to speed up the detection procedure. After back projection, the input image is then thresholded and filtered to be a binary image, and then a blob detection method named "adaptive blob growing" is executed on the whole binary image very quickly. Much information of original image is lost to be a binary image, and so the blob detection speed is increased a lot with the scarification of precision. The precision of position, scale and orientation is then recovered at the second stage by adaptive bandwidth mean shift [9]. The adaptive bandwidth mean shift is a real time algorithm for seeking optimum locally. Thus the whole detection algorithm is very fast without loss of precision.

Our approach is different from [10] and CAMSHIFT[11]. In Paper [10] and [11], mean shift iteration is executed directly on the back projected image to seek local optimum. But in our approach, back projection and blob detection procedure are only used to get initial rough positions globally. A second stage is executed to refine the detection results, and the adaptive bandwidth mean shift is applied to the original image instead of the back projected grayscale image. Compared with our previous approach ABMSOD in paper [2], the rough stage of our algorithm is the replacement of sliding-window method, and then the detection time is greatly reduced.

The remainder of this paper is organized as follows. Section II describes the details of the rough detection stage. Section III describes the precise detection stage using the adaptive bandwidth mean shift. Section IV is the experiments. A conclusion is given in Section V. Section VI is acknowledgements.

II. Rough Detection

The objective of rough detection is to detect the initial object locations quickly in the whole image. Rough detection is the replacement of sliding-window method, and it sacrifices precision for speed. The detection results are the inputs for the second precise detection stage. The precision of initial position values are not important at the first stage. It is enough for the second stage if the initial regions have some overlapped area with the candidate model.

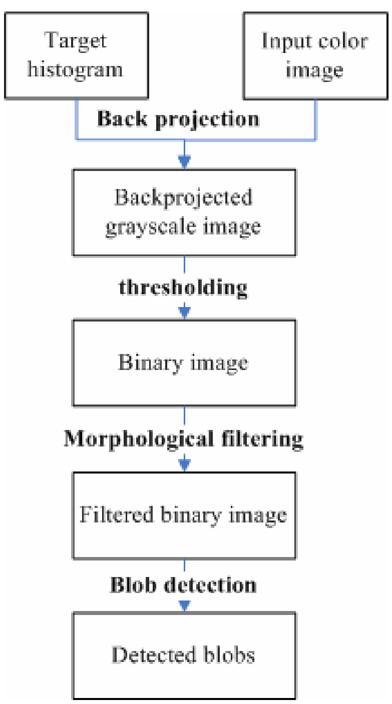

Fig. 1. Rough Detection Stage

There are four steps at the rough detection stage. First, back projection is introduced to produce a grayscale image. Back projection is a kind of saliency mechanism. And then thresholding is applied to avoid the minor colors, to reduce the saliency of dominant colors as well as to speed up the rough detection stage. Morphological operators are then applied to filter the obtained binary image to reduce noise and minor colors. Finally blobs in the filtered image are detected using "adaptive blob growing" method to complete the rough global optimum stage. After back projection, thresholding and filtering, most information of candidate objects is lost, and so the blob detection procedure can only get some parts of original objects and the precision is not very good for the sake of speed. But it is enough as initial inputs for adaptive bandwidth mean shift framework.

A. Back projection

In back projection, Histograms are back projected onto the input image. The histograms used are adaptive bandwidth kernel weighted color histograms which are illustrated in [9]. Back projection is to replace each pixel's color value with the height of the corresponding normalized histogram bin in the color space. The gray value of pixel (i, j) would be:

$$b(i,j) = 255 \times hist(bin(color_{i,j})) \tag{1}$$

After back projection, the image becomes a grayscale image. Pixels with dominant colors would have higher gray values while other pixels would have lower gray values. Back projection is one kind of visual attention mechanisms. The distances between candidate pixels and non-candidate pixels are enlarged by back projection. Pixels of the input image that have similar colors with the target model would have more chances to be parts of objects.

B. Thresholding

In order to detect objects quickly, we only pay attention to the dominant colors which are above some threshold. Thresholding is executed on the grayscale image after projection. Pixels whose gray values are above the threshold are treated equally as white pixels. The others are black pixels. Thresholding is a way to enable our algorithm to deal with histograms containing multiple dominant colors. The question is how to choose the threshold. If the threshold is small, more colors would be included. If the threshold is high, only a few colors would be included. In our application, the threshold t is:

$$t = 0.5 \times \max\_hist \tag{2}$$

C. Morphological filtering

After thresholding, the binary image is filtered against noise and small regions by morphological operators. In this paper, erosion and dilation morphological operators are used. At the beginning, erosion is used to remove the small regions, and then dilation is used to merge two nearby regions. A 3×3 rectangular structuring element is used for both dilation and erosion.

D. Blob detection

Blob detection is used to find regions in the filtered binary image. The initial positions and scales are then derived for the second stage.

Labeling algorithms are suitable for blob detection. In order to detect blobs quickly, an adaptive blob growing method is proposed. In the method, an unlabeled white pixel is first labeled as the center of a new region. And then the region grows in top, bottom, left and right directions if there are unlabeled white pixels which have 8-connectivity relationship with the region. The region grows until there are no more unlabeled white pixels connected to the region. The above procedure is iterated until all white pixels are labeled. The enclosing rectangles of the blob regions are then derived.

The algorithm is summarized below:

TABLE I
ADAPTIVE BLOB GROWING

```
Q := {all unlabeled white pixels}
current_label := -1;
OUTER_LOOP:
For ( ; size(Q) > 0 ; )
{
    current_label := current_label + 1;
    p := pop_first(Q);
    push( region[current_label], p);
   rectangle[current_label] inited as pixel p;
   growing_step := 1;
   DETECT:
   for each direction DIR in {left, right, top, bottom}
   {
      for each 8_connectivity(rectangle, growing_step )
     pixels q  of direction DIR in Q
      {
        q = remove_from(Q, q);
        push(region[current_label], q);
        rectangle[current_label].DIR := q.x or q.y
      }
   }
   if ( rectangle[current_label] in four directions is
    enlarged)
   {
      Growing_step := Growing_step * 2;
   }
   else if(rectangle([urrent_label] in four direction is not
   enlarged)
   {
       If(growing_step == 1)
      {
           if min_distance(current region, other regions) <
               min_threshold
           {
             Merge(region[current_label], other_region);
           }

          Goto OUTER_LOOP;
      }
      else
      {
         Growing_step := Growing_step / 2;
      }
   }
 Goto DETECT;
}
Discard small regions demagnetizing factor
```

The blob detection algorithm can detect all the blobs in the binary image. Large blobs are accepted, and others are rejected. If no blob is accepted, it is not necessary to execute the precise detection stage. The object detection algorithm reports no object existence. If there are accepted blobs, the detected blob rectangles are taken as initial inputs for the precise detection stage.

The four steps of rough detection are illustrated in Fig 2. After back projection, the contrast of object and background is enlarged a lot. However, there are still many noises. When thresholding is applied, pixels containing the unimportant colors in the original image are treated as background, but there are still many small regions and many holes in the image. These problems are solved by morphological filters. In the forth figure, only parts of the panda are reserved, there are two blobs detected from binary image in the fifth figure.

In all, back projection, thresholding, morphological filter and blob detection constitute the rough detection stage. The combined backprojection and thresholding method are suitable for efficiently detecting probable objects in several milliseconds. The precision of position and size of the detected blobs in the rough detection stage is not very good. The results from the rough detection stage are the initial inputs for the next stage for precision refinements.

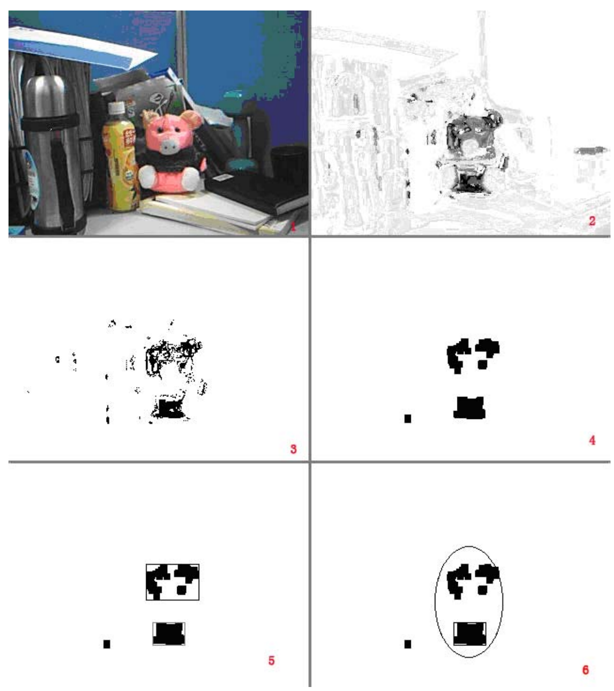

Fig. 2. Illustration of the four steps in rough detection. The first image is the input image. The second is the back projected image. The third is the binary image after thresholding. The fourth is the binary image after morphological filtering. The fifth is the detected blobs in the binary image. In the sixth image, the ellipse is the detected precise result from the corresponding rectangle. The grayscale image are **NOT**ed for easier viewing.

III. Precise Detection

Rough object detection stage is sufficient to identify the existence of objects and to get the rough position and size of objects quickly. However, it is not enough to get the precise size, position and orientation of objects. To deal with this problem, we applied adaptive bandwidth mean shift algorithm to detect these parameters of objects precisely.

At the precise detection stage, the HSV color model is the suggested color model, while other models can also be used. The adaptive bandwidth kernel weighted color histogram is used for target model and candidate model description. The Bhattacharyya coefficient is used for measuring similarity of two models. The adaptive bandwidth mean shift framework is applied to seek the local optimum, and then the precise size, position and orientation of objects are derived. Adaptive Bandwidth Mean Shift is our previous work. The detail of the algorithm is illustrated in paper [9]. The algorithm is briefly introduced in the following.

A. Adaptive Bandwidth Mean Shift Frame work

Mean shift was first proposed in 1975 [12]. Since Cheng's work in 1995 [13], many researchers paid much attention to the mean shift algorithm. Comaniciu has done lots of work to apply the mean shift algorithm to image segmentation [14], object tracking [15], object detection [1], etc. and has achieved great success. Mean shift is one of the best algorithms for seeking local optima in real-time usage. There are two key problems in the conventional mean shift algorithm: the first is how to select the initial bandwidth parameters, and the second is how to adjust the bandwidth according to data changes. Comaniciu proposed a data-driven method for bandwidth adjustment [16]. However it is complicated.

We have generalized the conventional mean shift algorithm to be the Adaptive Bandwidth Mean Shift algorithm [9] to solve the above two problems. In the adaptive bandwidth mean shift algorithm, a symmetric bandwidth matrix instead of a diagonal bandwidth matrix is used. We have proved that when the bandwidth matrix is definite positive, the mean shift vector will point to the direction where the kernel density function increases. The bandwidth matrix can be used for description of a super ellipse ball region where samples are located. Initially, the bandwidth matrix can be calculated according to the initial region of sample points. Each mean shift iteration contains a mean shift step and a bandwidth matrix optimization step, and it is executed iteratively until convergence. Then the position and the optimal bandwidth matrix are calculated. According to the optimal bandwidth matrix, we can calculate the target sample regions after convergence. In 2D case, the bandwidth matrix describes an ellipse, which represents object orientation and scale.

B. The Precision Stage

The adaptive bandwidth mean shift is applied for seeking the local optimum of similarity between the candidate model and the target model. The precise size, position and orientation of objects can be obtained consequently. The corresponding algorithm is given in the following table.

TABLE II
ABMS for local optimum

Given the target model $\{\hat{q}_u\}_{u=1...M}$.
For each detected blob from the rough detection stage, do:
1) Calculate the initial bandwidth matrix $\boldsymbol{H}_0$ and position $\boldsymbol{y}_0$ according to S_0 , which is the initial rectangle.
2) Derive the weights $\{w(\boldsymbol{s})\}_{s\in S_0}$.
3) Execute one mean shift iteration to find the next location $\boldsymbol{y}_1$
4) Move the window center to $\boldsymbol{y}_1$, update S_0 recalculate $\{w(\boldsymbol{s})\}_{s\in S_0}$.
5) Update the bandwidth matrix
6) recalculate region S_1 .
7) if the points in S_1 and S_0 are the same, goto 9); Else $\boldsymbol{y}_0 \leftarrow \boldsymbol{y}_1, S_0 \leftarrow S_1, \boldsymbol{H}_0 \leftarrow \boldsymbol{H}_1$, goto 2).
8) Calculate the kernel weighted histograms in region S_0 , calculate the similarity with the target model, if the similarity is below the threshold, the blob is refused; else the blob is accepted, position $\boldsymbol{y}_1$, orientation ϕ , two half lengths of both axes $\sigma\hat{a}$, $\sigma\hat{b}$ are obtained.

If the inputs are image sequences, there is no need to detect objects in every image. Instead, we just run the whole detection algorithm when it is the first image. We run the ABMS algorithm for tracking the local optimum until failure.

C. The Whole Object Detection Algorithm

The overall object detection algorithm can be written as below:

TABLE III
Rapid and precise object detection algorithm using ABMS

A Target model building
1)Given a target image with initial region $\hat{S}_0$ (ellipse) centered at position $\hat{\boldsymbol{x}}_0$ with two half axis lengths $a_0 = \sigma\hat{a}_0$, $b_0 = \sigma\hat{b}_0$ and its initial orientation $\hat{\phi}_0$, the initial bandwidth matrix $\hat{\boldsymbol{H}}_0$ can be initialized
2)calculate the normalized kernel weighted color histogram of target model.
B Object detection
Given an image and target model
1) Run the rough object detection stage to detect probable blobs. If there is no blob, report failure, stop.
2) Run the precise object detection stage to get the precise size, position and orientation. If the final similarity is lower than similarity threshold, report failure.
Given image sequences and target model
For each image:
1) If it is the first image or it is failure in previous image, run the rough object detection stage.
2) Run the precise object detection stage. If success, reports the results, stop. If failure, go to 3)
3) If a rough object detection stage has already been executed, reports failure, stop.
4) Run both of the rough object detection stage and the precise object detection stage, report results. Stop.

IV. EXPERIMENTS

Two experiments are designed to verify the object detection ability. The first experiment is used for speed verification. The second is used for precision verification.

A. The Speed Experiment

To verify the speed performance of the algorithm, we have compared our algorithm with integral histogram based detection algorithm [6] and SIFT detection algorithm[17]. CAMSHIFT [11] is not included for comparison because it is usually used for seeking local optimum for object tracking applications. What's more, it is only proper for objects with single dominant color. The comparison of adaptive bandwidth mean shift and classical mean shift has already been done in [9].

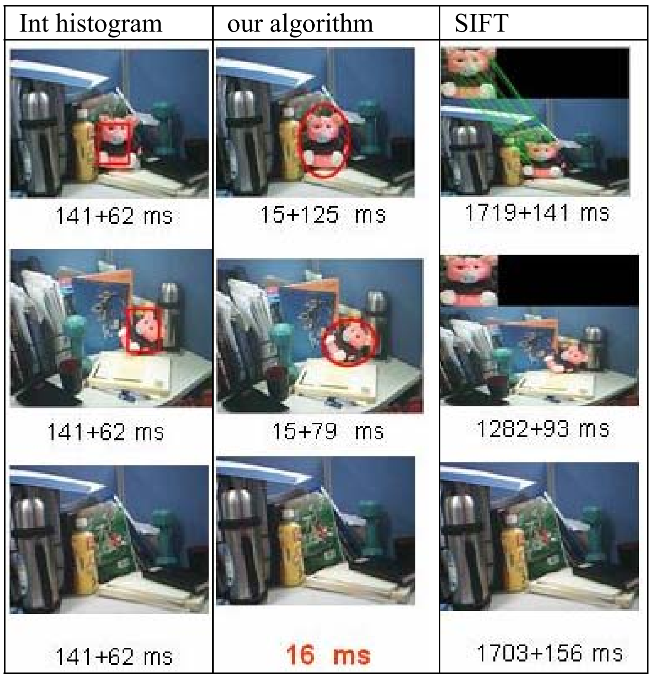

Fig. 3. Comparison of our algorithm with integral histogram based detection algorithm and SIFT detection algorithm. our algorithm is much faster than the other two algorithms.

The three algorithms are tested with the same inputs under the same conditions. The size of input color images is 320×240. The three algorithms run on the same PC with the configuration of P4 CPU and 512MB memory. HSV color model of 32×32×1 bins is used for histogram building. Integral histogram and use the same color model.

Three images are used for detection. They are in the first, second, and the third row in Fig 3. There are two numbers indicating times needed for each image in Fig3. For integral histogram based algorithm, they are time for calculation of integral histogram and global optimum. For our algorithm, they are time for rough detection and precise detection. For SIFT, they are time for feature extraction and matching. In the integral histogram algorithm, the total time needed is 203 ms, and it consumes the same time regardless of the existence of objects. In our algorithm, that larger the object is, more time

Fig. 4. Detection results of bottles and toy pandas of different scales and orientations in six images.

is needed for detection by our algorithm. When there is no object, the time consumed is only 16 ms. The average time needed is 83ms. SIFT is much slower than color histogram base algorithms. Compared with ABMSOD [2], our algorithm has improved a lot in speed. The average time consumed for each image of ABMSOD is 346ms, but the time consumed in our algorithm is 140ms at most, the average time consumed in our algorithm is 83ms.

B. The Precision Experiment

The second experiment in Fig 4 shows the detection results by our algorithm in six images. Toy pandas and drink bottles of different scales and orientations were successfully detected. More detailed detection error could be found in paper [9].

From the above experiments and analysis, we can draw the conclusion that color histograms based algorithm are much faster than non color histograms based object detection algorithms. Our algorithm is faster than the integral histogram based approach and our previous ABMSOD approach. The precision of our algorithm is much better than integral histogram and SIFT. Our algorithm is able to detect objects rapidly and precisely.

V. Conclusion

In this paper, we have proposed a rapid and precise color histogram based object detection algorithm using adaptive bandwidth mean shift. The major contribution of our work is that we have proposed a novel rough detection procedure to replace the sliding-window method for fast object detection. The rough detection stage is able to reach global optimum in several milliseconds, which dramatically improves detection speed. Because the precision of the rough detection is sacrificed, the rough detection stage must be accompanied with adaptive bandwidth mean shift to detecting objects precisely.

Our algorithm has many advantages. First, it is a general object detection algorithm which runs very fast; Second, it is suitable for models that have multiple dominant colors; third, when there is no object, global optimum needs much less time; forth, position, size, and orientation can be simultaneously and precisely detected and tracked; last, local optimum instead of global optimum can be applied to image sequences for faster detection. In all, this is a general color histogram based object detection algorithm. It is very suitable for applications which have critical requirements in speed and precision.

There are several thresholds that we should pay attention to. They are the histogram threshold for thresholding the back projected image, the blob area threshold for accepting blobs and the threshold for similarity of candidate model and target model. They are all selected experimentally. What we plan to do in the future is to develop a method for automatically choosing these thresholds.

The blob finding method may be improved for speeding up the rough detection stage in the future. In our algorithm, color is used as a local feature for every pixel. We are going to try other features such as textures and edges for object detection in the same algorithm framework.

VI. Acknowledgements

Many thanks to professor Hong Zhang (He is with the department of Computing Science, University of Alberta, email: zhang@cs.ualberta.ca) for his helpful suggestions on revision of the paper.

References

[1] D. Comaniciu, V. Ramesh, "Robust detection and tracking of human faces with an active camera," in *Third IEEE International Workshop on Visual Surveillance*, 2000, pp. 11-18.

[2] X. Chen, H. Huang, H. Zheng, et al., "Adaptive Bandwidth Mean Shift Object Detection," in *IEEE Conference on Robotics, Automation and Mechatronics*, Chengdu, 2008, pp. 210-215.

[3] M. J. Swain, D. H. Ballard, "Color indexing," *International Journal of Computer Vision,* vol. 7, pp. 11-32, 1991.

[4] D. Comaniciu, P. Meer, "Robust Analysis of Feature Spaces: Color Image Segmentation," 1997, pp. 750-755.

[5] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-Based Object Tracking," *Ieee Transactions on Pattern Analysis and Machine Intelligence,* pp. 564-577, 2003.

[6] F. Porikli, "Integral Histogram: A Fast Way To Extract Histograms in Cartesian Spaces," in *IEEE Computer Society Conference On Computer Vision and Pattern Recognition*, 2005, p. 829.

[7] R. T. Collins, "Mean-shift blob tracking through scale space," 2003.

[8] Z. Zivkovic, B. Krose, "An EM-like algorithm for color-histogram-based object tracking," 2004.

[9] X. Chen, Y. Zhou, X. Huang, et al., "Adaptive Bandwidth Mean Shift Object Tracking," in *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, Chengdu, 2008, pp. 1011-1017.

[10] G. Jaffre, A. Crouzil, "Non-rigid object localization from color model using mean shift," in *2003 International Conference on Image Processing*, 2003, pp. 317-320.

[11] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *IEEE Workshop on Applications of Computer Vision*, 1998, pp. 214–219.

[12] K. Fukunaga, L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Information Theory, IEEE Transactions on,* vol. 21, pp. 32-40, 1975.

[13] Y. Cheng, "Mean Shift, Mode Seeking, and Clustering," *Ieee Transactions on Pattern Analysis and Machine Intelligence,* p. 790799, 1995.

[14] D. Comaniciu, P. Meer, "Robust Analysis of Feature Spaces: Color Image Segmentation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 750-755.

[15] D. Comaniciu, V. Ramesh, P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[16] D. Comaniciu, V. Ramesh, P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Eighth IEEE International Conference on Computer Vision*, 2001.

[17] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, pp. 91-110, 2004.