

# Robust Visual Tracking via Fuzzy Kernel Representation

Zhiqiang Wen

School of Computer and Communication, Hunan University of Technology, Zhuzhou, Hunan, 412008, China  
zhqwen20001@163.com

Yongxin Long

School of Computer and Communication, Hunan University of Technology, Zhuzhou, Hunan, 412008, China  
53767768@qq.com

Zhaoyi Peng

School of Computer and Communication, Hunan University of Technology, Zhuzhou, Hunan, 412008, China  
pengzhaoyi@163.com

**Abstract**—A robust visual kernel tracking approach is presented for solving the problem of existing background pixels in object model. At first, after definition of fuzzy set on image is given, a fuzzy factor is embedded into object model to form the fuzzy kernel representation. Secondly, a fuzzy membership functions are generated by center-surround approach and log likelihood ratio of feature distributions. Thirdly, details about fuzzy kernel tracking algorithm is provided. After that, methods of parameter selection and performance evaluation for tracking algorithm are proposed. At last, a mass of experimental results are done to show our method can reduce the influence of the incomplete representation of object model via integrating both color features and background features.

**Index Terms**—Fuzzy Kernel Histogram, Visual Tracking, Fuzzy Factor, Fuzzy Membership Function

## I. INTRODUCTION

Since mean shift and its application appeared in 1999 [1], it has been widely used for object tracking, image segmentation, pattern recognition, clustering, filtering, etc. Comaniciu, et al [2] firstly used mean shift algorithm to track moving object. He regarded Bhattacharyya coefficient as the comparability measurement between object model and an object candidate and used mean shift algorithm to find the optimum object candidate. Peng, et al [3] proposed the automatic selection of bandwidth for mean shift object tracking. However, there are two factors which will affect the performance of object tracking. One factor is background pixels in object model. Background pixels in object model will increase errors of tracking, but in order to let the object being contained in object model, it is inevitable to introduce some background pixels in object model. For resolving this problem, a simple

approach is to omit the background pixels from object model. This method is robust against disturbance of background pixels in object model, but there are many difficulties for it. There are other methods about how to omit the background pixels, for example Collins, et al [4] used a center-surround approach to sample pixels from object model and the log likelihood ratio of these sample pixels as a new feature was used to represent the object model in kernel tracking. Feng et al [5] presented an image matching similarity criterion based on maximum posterior probability via the statistical feature of searching region to reduce the background pixels. Another factor is insufficient character information. Traditionally kernel histogram is used to describe the color space statistical distribution of object for the reason that it is invariant to rotation and translation of image, but kernel histogram has an insufficient space information issue. Therefore, it needs to find a simple and feasible method of describing both image space information and color information. For example, Zhu, et al [6] used the geometry global shape context information to improve the description of object. Zivkovic, et al [7] used the color histogram of 5-degrees of freedom to track the revolving object. Other methods are as spatial color histogram [8], mixed model of dynamic texture [9], etc.

In this paper, a robust kernel tracking approach is presented to reduce errors of visual tracking. In our tracking, a fuzzy factor is embedded into object model to form the fuzzy kernel histogram which is different from fuzzy color histogram [10] in which FCM clustering technique was used to solve inherent boundary issue. In our fuzzy object representation, the background pixels in object model are used to build fuzzy membership function for improving accuracy of kernel tracking.

## II. FUZZY KERNEL REPRESENTATION

In this section, a method of building fuzzy object representation is presented by combing both kernel histogram and fuzzy set. A fuzzy set is defined as follows.

Manuscript received Mar. 1, 2012; revised Apr. 1, 2012; accepted Apr. 12, 2012.

Project number: 61170102, 11JJ3070

DEFINITION 1. If  $\mathbf{A}$  denote the set of object pixel and universe  $\mathbf{U}$  represent the color features in object model, a fuzzy set  $A$  on universe  $\mathbf{U}$  is defined as  $\mathbf{A} = \{\mu_A(\mathbf{x}), \mathbf{x} \in \mathbf{U}\}$ , where  $\mu_A(\mathbf{x}) \in [0, 1]$  named as membership of set  $\mathbf{A}$  on  $\mathbf{U}$ .

In above DEFINITION,  $\mu_A(\mathbf{x})=0$  ( $\mu_A(\mathbf{x})=1$ ) means membership is not attributed (completely attributed) to  $\mathbf{A}$  on  $\mathbf{U}$  and  $\mu_A(\mathbf{x})=0.5$  represents the fuzzy boundary point of set  $\mathbf{A}$  on  $\mathbf{U}$ . According to DEFINITION 1,  $\mu(\mathbf{x})$  shows the degree that color in object model is attributed to object, so a fuzzy membership factor is introduced in kernel object representation and candidate object respectively as follows.

$$p_u(\mathbf{y}) = C_H \mu_u(\mathbf{y}) \sum_{i=1}^{n_H} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{\mathbf{H}}\right\|^2\right) \delta[b(\mathbf{x}_i^*) - u] \quad (1)$$

$$q_u = C \mu_u \sum_{i=1}^n k\left(\left\|\mathbf{x}_i^*\right\|^2\right) \delta[b(\mathbf{x}_i^*) - u] \quad (2)$$

In (1) and (2), both  $\mathbf{q} = \{q_u\}$  and  $\mathbf{p}(\mathbf{y}) = \{p_u(\mathbf{y})\}_{u=1, \dots, m}$  are object model and object candidate respectively where  $\mathbf{y}$  is the center location of object candidate, satisfying  $\sum q_u = 1$  and  $\sum p_u = 1$ . In addition,  $m$  is the number of bins and  $\delta(\bullet)$  is the Kronecker delta function.  $\{\mathbf{x}_i^*\}_{i=1, \dots, n}$  is the normalized pixel locations in object region which is centered at 0. Function  $b: R^2 \rightarrow \{1, \dots, m\}$  associates to the pixel at location  $\mathbf{x}_i^*$  the index  $b(\mathbf{x}_i^*)$  of its bin in the quantized feature space. Both  $C$  and  $C_H$  are the normalization constant. Both  $n$  and  $n_H$  are the total pixel number in object region.  $k(\cdot)$  is a kernel function and  $\mathbf{H}$  is the bandwidth matrix. For (1) and (2), the key problem is how to acquire  $\mu_u$ . Membership function of fuzzy set on universe  $\mathbf{U}$  is actually the real function that  $\mathbf{x} \in \mathbf{U}$  is mapped to  $[0, 1]$ . Moreover, fuzzy set reflect the subjectivity of human brain against the external thing, so the fuzzy membership function is complicated and diversiform. Generally, there are three methods for fuzzy membership function, namely fuzzy statistics, fuzzy dual contrast compositor method and compound weight method. Since universe  $\mathbf{U}$  is in real number field, for fuzzy set  $\mathbf{A}$  in  $\mathbf{R}$ , fuzzy membership function  $\mu_A(\mathbf{x})$  is named as fuzzy distribution. There are many forms about fuzzy distribution such as Normal distribution, Cauchy distribution, etc. In next section, we will introduce how to acquire fuzzy membership function for factor  $\mu_u$ .

### III. FUZZY MEMBERSHIP FUNCTION

In kernel tracking, according to the definition of fuzzy set, membership factor  $\mu_u$  should show the degree that kernel histogram features attribute to the object, so  $\mu_u=0$  means the  $u$ th background feature is similar to corresponding object feature which should be omitted.  $\mu_u=1$  shows the  $u$ th background feature is different from corresponding object feature which should be reserved. According to above, fuzzy dual contrast compositor method is used to get fuzzy membership, namely  $\mu_u$  should be computed via both object model and background pixel. Basic process of creating fuzzy kernel histogram in this paper is shown in Figure 1. In Figure 1,

after kernel histogram of object color and the histogram of background color are extracted respectively from the video frame named  $g_u$  and  $b_u$ , a fuzzy membership function is used to form fuzzy membership factor  $\mu_u$  by utilizing both  $g_u$  and  $b_u$ . Then, the fuzzy kernel histogram is created by both (1) and (2). In this paper we use the center-surround approach[4], which is an effective method to extract the background feature and object feature, to form fuzzy membership function. In this method, after two regions (namely object region and background region) are built, the sampled pixels are acquired from these regions to establish color histograms. As described as Figure 2, firstly a rectangular set of pixels covering the object is chosen to represent the object pixels, while a larger surrounding ring of pixels is chosen to describe the background. The inner rectangular set hypothetically contains  $h \times w$  pixels and the outer rectangular set is  $rh \times rw$  size, where  $r > 1$ . In this paper  $r=2$ . The object pixels and background pixels are acquired respectively to create color kernel histogram and color histogram both of which are respectively normalized to form the feature probability distribution  $g_u$  and  $b_u$ , where  $1 \leq u \leq m$ . After acquiring  $g_u$  and  $b_u$ , the next step is to decide the fuzzy membership function according to both  $g_u$  and  $b_u$ . A ratio strategy will be given to create the fuzzy membership functions in next paragraph.

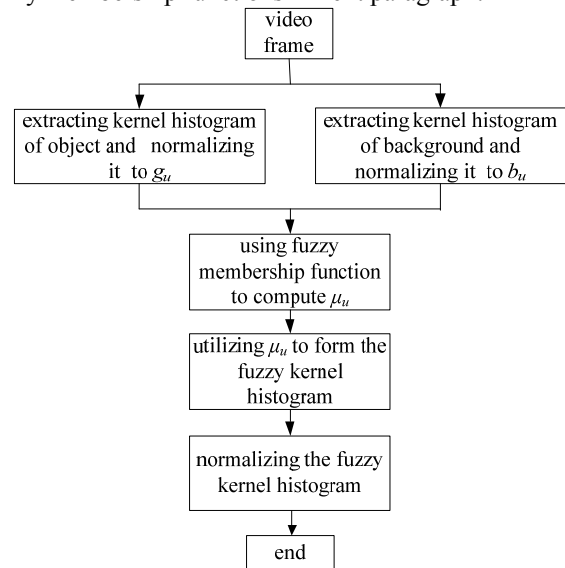


Figure1. Process of creating fuzzy kernel histogram



Figure2. Center-surround Approach

We present a fuzzy membership function as shown in (3), which is piecewise function of linear transformation. In (3), both  $B_1$  and  $B_2$  are positive constants limiting distribution of  $\mu_u$  in  $[0, 1]$  and  $L(u)$  is a new feature. The  $L$ - $\mu$  curve about (3) is shown in Figure 3. In Figure 3,  $\mu_u=0$ , if  $L(u) \leq -B_1$  while  $\mu_u=1$  if  $L(u) \geq B_2$ . For both object and background feature, a mapping function  $f(g_u, b_u)$  is used to form the new feature  $L(u)$  for better

discrimination between object and background, namely  $L(u) = f(g_u, b_u)$ . But how to select function  $f(g_u, b_u)$ ? It requires that  $L(u)$  must describe no background feature, but the object feature in object model. That is to say, large  $b_u$  describes the background pixels, and  $L(u)$  is a minimum value. On the other hand, small  $b_u$  shows  $g_u$  is the object feature which should be reserved. According to above,  $f(g_u, b_u)$  is decided by the log likelihood ratio of the feature distributions as shown in (4).

$$\mu_u = \max(0, \min(L(u) + B_1, B_1 + B_2)) / (B_1 + B_2) \quad (3)$$

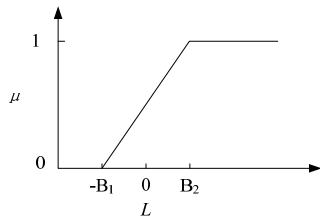


Figure3.  $L-\mu$  curve

$$f(g_u, b_u) = \log \frac{\max\{g_u, \delta\}}{\max\{b_u, \delta\}} \quad (4)$$

where  $\delta$  is a small value ( $\delta=0.1 \times 10^{-10}$  in this paper) to avoid the divisor being zero or the function value being negative infinite. In (4), the effects of background pixels in object model will be omitted. When the tracking localization is accurate or there are no occlusion, mean shift is good for object tracking by only using feature  $L(u)$ . In this case, the representation of object model and background information is correct, but if occlusion occur in other scene, the incomplete representation of object model will result in error of  $L(u)$  and will enlarge the errors of object tracking. Fuzzy kernel histogram integrates both color features and background features to reduce the influence of the incomplete representation of object model.

IV. TRACKING BASED ON FUZZY KERNEL HISTOGRAM

In kernel tracking, Bhattacharyya between  $\mathbf{p}(\mathbf{y})$  and  $\mathbf{q}$  is  $\rho(\mathbf{y}) = \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}]$ . After using Taylor expansion on Bhattacharyya coefficient between object model and object candidate, we can get the next location by mean shift as

$$\mathbf{y}_{k+1} = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i g\left(\left\|\frac{\mathbf{y}_k - \mathbf{x}_i}{\mathbf{H}}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\mathbf{y}_k - \mathbf{x}_i}{\mathbf{H}}\right\|^2\right)} \quad (5)$$

where

$$w_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(\mathbf{y}_k)}} \delta[b(\mathbf{x}_i) - u]$$

and  $g(\cdot) = -k^2(\cdot)$ . So the fuzzy kernel tracking (FKT) algorithm for finding object position in  $t$ th frame can be described as follows.

Step 1. Initialize the object location  $\mathbf{y}_0$  in the current frame and other parameters  $\mathbf{H}$ ,  $m$ ,  $B_1$ ,  $B_2$ , small real number  $\epsilon$ ,  $k=0$ , maximum iterations  $N$ , etc.

Step 2. Compute  $q_u$  according to (2) in the sample frame.

Step 3. Compute  $p_u(\mathbf{y}_{k+1})$  according to (1) in next frame, then calculate weight  $w_i$ .

Step 4. Find the next location  $\mathbf{y}'_{k+1}$  of the object candidate according to (5).

Step 5. Correct the location according to the moving information of object, namely  $\mathbf{y}_{k+1} = \mathbf{y}'_{k+1} + \Delta t \mathbf{v}_k$ , where  $\mathbf{v}_k = (\mathbf{y}'_{k+1} - \mathbf{y}_k) / \Delta k$ . For the simplicity, let  $\Delta k = 1$ .

Step 6. If  $\|\mathbf{y}_{k+1} - \mathbf{y}_k\| < \epsilon$  or  $k < N$ , stop algorithm, else let  $k=k+1$  and go to Step 3.

The output value  $\mathbf{y}_k$  in FKT algorithm is regarded as the object position  $\mathbf{y}_t$  in  $t$ th frame. Parameter  $k$  describes the iterations of tracking algorithm. Moreover, in Step 1, popularly  $\mathbf{y}_0$  is set as the position of object in sample frame. The effect of step 5 is that effective tracking results can be acquired by linear prediction for the new position of object in next iteration.

V. PARAMETER SELECTION

For above fuzzy kernel tracking algorithm, there are some other problems in object tracking.

A. Selection of Kernel.

Three kernels frequently used in object tracking are Gaussian kernel  $k(x) = e^{-x}$ , Epanechnikov kernel

$$k(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1-x) & \text{if } x \leq 1 \\ 0 & \text{else} \end{cases}$$

and Truncated Gaussian kernel

$$k(x) = \begin{cases} e^{-x} & |x| \leq \lambda \\ 0 & |x| > \lambda \end{cases}$$

There are three reasons for selecting truncated Gaussian kernels. (1) For Gaussian kernel, when  $x$  is very large, the value of Gaussian kernel is too small and has little influence on object tracking, but it will increase the computing time. (2) For Epanechnikov kernel, mean shift will turn to

$$\mathbf{y}_1 = \sum_{i=1}^{n_h} \mathbf{x}_i w_i / \sum_{i=1}^{n_h} w_i$$

which isn't proved convergent. (3) For truncated Gaussian kernel, the computing time will be reduced and its convergence has been proved [11].

B. Bandwidth Matrix.

Using fully parameterized  $\mathbf{H}$  increases the complexity of the estimation and in practice,  $\mathbf{H}$  is chosen either as  $\mathbf{H} = \text{diag}[h^2_1, \dots, h^2_d]$  or proportional to the identity matrix  $\mathbf{H} = h^2 \mathbf{I}$ . Let  $h_{prev}$  is the bandwidth in previous frame and there are three bandwidth values as  $h = h_{prev}$ ,  $h = h_{prev} + \Delta h$  and  $h = h_{prev} - \Delta h$  where  $\Delta h = 0.01 h_{prev}$ . A simple method [2] is to find optimal bandwidth  $h_{opt}$  possessing the maximum Bhattacharyya coefficient by mean shift. The fault of this method is that mean shift will run three times and the compute time will also increase. In our method, there are three parameters about object: the center position vector  $\mathbf{y}$ , width and height. For the simplicity,  $\mathbf{H} = \text{diag}[h_x, h_y]$ , where  $h_x$  and  $h_y$  are the width and height of object respectively. After the object center is found by FKT algorithm, optimum  $h_x$  and  $h_y$  are searched locally, so that Bhattacharyya coefficient  $\rho(\mathbf{y})$  reaches a maximum. The

search region cannot be set too large, otherwise, the real-time is bad.

C. Selection of Initial Location.

Generally the location of object in previous frame is regarded as the initial point of FKT iteration in the current frame. In this way, if object move slowly, it is good for object tracking, but if object move quickly or occlusion occurs, mean shift could not find the object. The reason is that the similar function is expanded by Taylor formula around  $p_i(\mathbf{y}_0)$  which means the distance between  $\mathbf{y}_0$  and  $\mathbf{y}$  should not be too large. We can predict the object location as the initial location of FKT iteration by analyzing a series of object location in former frames according to the property of inertia and continuity of moving object. There are many predictive methods such as Kalman filter, etc. In order to ensure the real-time performance of tracking method, a linear prediction method is used in this paper. The method is  $\mathbf{y}_{t+1} = \mathbf{y}_t + \Delta t \cdot \mathbf{v}_t$ , where  $\Delta t$  is the time interval between two frames and  $\mathbf{v}_t$  is the movement speed of object between  $t$ th and  $t+1$ th. Speed  $\mathbf{v}_t$  can be computed by formula  $\mathbf{v}_t = (\mathbf{y}_{t+1} - \mathbf{y}_t) / \Delta t$ . for simplicity and the equal time interval, let  $\Delta t = 1$ . Furthermore, in (3), let  $B_1 = \log(5.0)$  and  $B_2 = \log(2.0)$ , we can get good result.

VI. EXPERIMENTAL RESULTS

A. Experiment Conditions

A dataset composed of 9 different tracking video is used to test our tracker validity. In this dataset, there are three category objects which are respectively head target ( $H_1, H_2, H_3,$  and  $H_4$ ), pedestrians ( $P_1, P_2, P_3,$  and  $P_4$ ) and vehicle ( $V_1$ ).  $H_1, H_2, H_3,$  and  $H_4$  come from the public dataset<sup>1</sup>.  $P_1$  is a part of PETS2001 dataset<sup>2</sup>, and  $P_2$  is from CAVIAR dataset<sup>3</sup>. Other videos are that we collect from monitoring system in diverse scenes. Figure 4 shows the sample frames of dataset. There are some characteristic about moving object on dataset. In  $H_1, H_2, H_3$  and  $H_4$ , the velocity of head movement is quick and sometimes changes suddenly. There are shields in  $H_4$ . In  $P_2$ , the scene is intricate and daylight illumination may vary. In  $P_1, P_3$  and  $P_4$ , the objects are moving slowly and there is little disturbance of daylight illumination. In  $V_1$ , velocity of the car is quick. In this paper, we use VC++6.0 and Open CV as programming tool and assume both the object and its initial location have been known.

B. Performance Evaluation

In order to compare the performance of existing tracking methods, tracking error estimation is used in this paper. Tracking error is defined as the deviation between estimating parameters of tracking method and real parameters. We define tracking error of  $t$ th frame as:  $error_t = \|\mathbf{x}_t - \mathbf{y}_t\|_2$  (pixels), where  $\mathbf{y}_t(\mathbf{y}_t^i, i=1, 2)$  is the parameter vector of object in  $t$ th acquired by tracking method and  $\mathbf{x}_t(\mathbf{x}_t^i, i=1, 2)$  is the real parameter vector of

object in  $t$ th. In our experiments, real parameter vectors of dataset  $H_1, H_2, H_3, H_4$  and  $P_1$  are come from the data provided by E. Maggio<sup>4</sup>, while the real parameter vector of other datasets is obtained via manual method. In addition, we define the average error of  $error_t$  as *error* in some sequential frames.

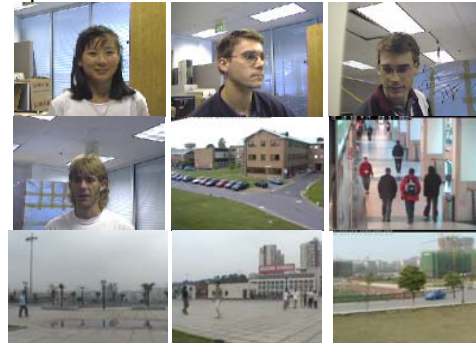


Figure 4. Dataset, (from top-left to bottom-right) Head object ( $H_1, H_2, H_3, H_4$ ); Pedestrians ( $P_1, P_2, P_3, P_4$ ); Vehicle ( $V_1$ )

C. Experiment Results

The experiment results of our method (FKT) and mean shift (MS)[2] on  $H_1$  video are showed respectively in Figure 5, Figure 6 and Figure 7. Figure 5 shows the results in 300<sup>th</sup>, 312<sup>th</sup>, 330<sup>th</sup>, 342<sup>nd</sup>, 375<sup>th</sup>, 388<sup>th</sup> frame respectively where the object in white rectangle (FKT) is much more accurate than that in black rectangle (MS), that is to say, the presented method is more effective than MS. The errors of object tracking in each frame are shown in Figure 6 which shows that in the vicinity of 53<sup>rd</sup>, 94<sup>th</sup> frame, the error of MS is larger than that of the FKT, while in other frames, distinction of the errors between MS and FKT is not obvious. The average Bhattacharyya coefficient value varies little in FKT comparing to MS as shown in Figure 7. From Figure 11, we know that the average error of FKT is less than that of MS. In the same way, more experiments on  $P_3$  are showed in Figure 8 and Figure 9.

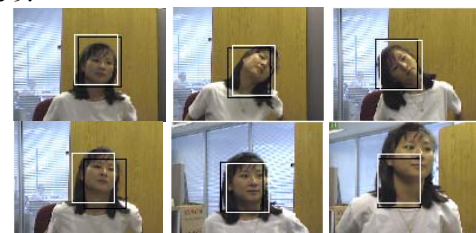


Figure 5. Experiment results of FKT and MS on  $H_1$  video. Above figures show the results in 300<sup>th</sup>, 312<sup>th</sup>, 330<sup>th</sup>, 342<sup>nd</sup>, 375<sup>th</sup>, 388<sup>th</sup> frame respectively, where black rectangle denotes MS and white rectangle denotes FKT.

Figure 10 shows a demonstration about fuzzy factor  $\mu$  in the first frame of  $H_1$  video. In Figure 10,  $\mu_u$  satisfies  $0 \leq \mu_u \leq 1$ .  $g_u \gg b_u$  (in experiment,  $g_u > 2 * b_u$ ) means  $\mu_u = 1$ . That is to say,  $g_u$  is the object color and should be kept.  $b_u \gg g_u$  (in experiment,  $b_u > 5 * g_u$ ) shows  $\mu_u = 0$ . This means that  $g_u$  is background color and should be omitted. Figure 11 shows the average error of MS and FKT respectively. It is obvious that the average error of FKT is lower than that of MS on video  $H_1, H_2, P_1, P_2, P_3, V_1$  and  $V_2$ , but larger than that of MS on other videos. According to the

<sup>1</sup> <http://www.ces.clemson.edu/~stb/research/headtracker/seq/>

<sup>2</sup> <ftp://ftp.pets.rdg.ac.uk/pub/PETS2001/>

<sup>3</sup> <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>

<sup>4</sup> <http://www.eecs.qmul.ac.uk/~andrea/multi-feature.html>

characteristic about moving object in dataset, it can be concluded that the FKT is adaptive to the scenes where object move slowing or there is little disturbance of daylight illumination. The reason of reducing errors is that in the fuzzy color kernel histogram, the effect of background pixels is omitted, and the good discrimination between object and background lead to more accurate localization. These results show our method is better. But in the scenes where there is serious disturbance of daylight illumination or shields, the accurate localization of FKT is less, for fuzzy color kernel histogram is impressible to varying background color.

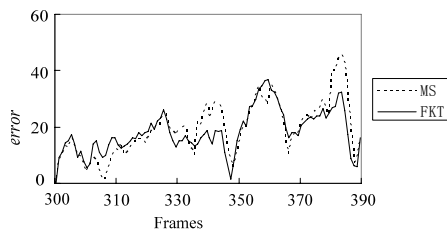


Figure 6. Error of Object Location in Some Frames

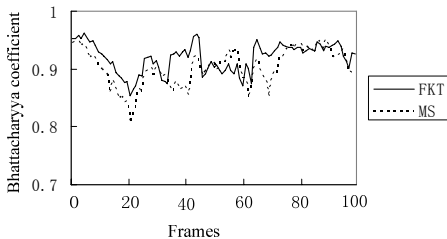


Figure 7. Comparison of Bhattacharyya Coefficient Value

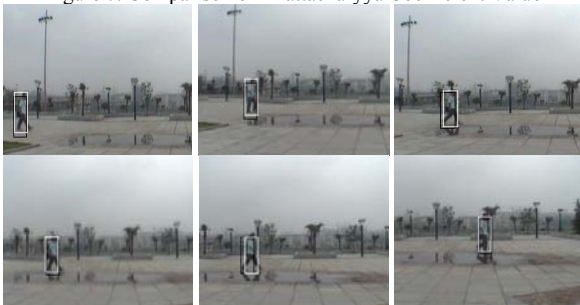


Figure 8. Experiment results of FKT and MS on P3 video above figures show the results in 4<sup>th</sup>, 29<sup>th</sup>, 42<sup>nd</sup>, 58<sup>th</sup>, 68<sup>th</sup>, 98<sup>th</sup> frame respectively, where black rectangle denotes MS and white rectangle denotes tracking results of FKT.

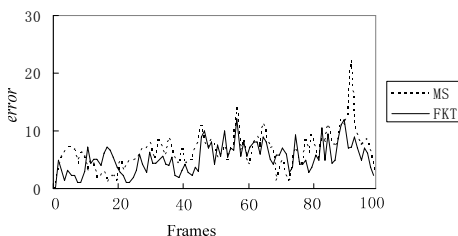


Figure 9. Error of object location in each frame.

**D. Effect of Parameter  $\lambda$**

Parameter  $\lambda$  will have an impact on performance as positioning accuracy, iteration times and runtime. So in this section, we will discuss how to select parameter  $\lambda$  by experiments. In our experiments, the tracking results is achieved by our algorithm respectively for  $\lambda \in [1,4]$  as shown in Figure12. Figure12 (a) shows the average error

with different  $\lambda$ . Figure 12(b) and Figure 12(c) show the iteration times and average time (Intel® core™ 2 DUO CPU 2.19GHZ and memory 2.98GB) respectively with  $\lambda \in [1,4]$ . From Figure12, we can conclude that small  $\lambda$  will result in bad position accuracy, little iteration times and quick computation speed but large  $\lambda$  will lead to bad real-time performance. Furthermore,  $\lambda$  is important for the performance of object tracking against occlusion. If  $\lambda$  is too small, the object is easily lost. If  $\lambda$  is too large, the computed time is too long and real-time performance is bad. Generally  $\lambda=1.5-2.5$ .  $\lambda=2$  in our experiment. Figure 13 (Figure 14) show the tracking result of FKT when  $\lambda=2$  ( $\lambda=1$ ) on video P<sub>4</sub> from which we can conclude that  $\lambda=2$  can provide a good ability against occlusion but  $\lambda=1$  will lead to object missed.

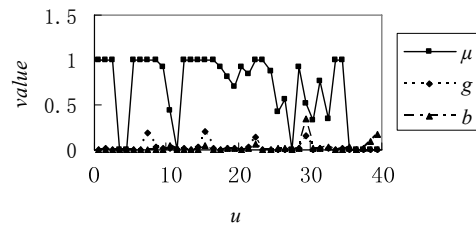


Figure 10. A demonstration about Fuzzy Factor

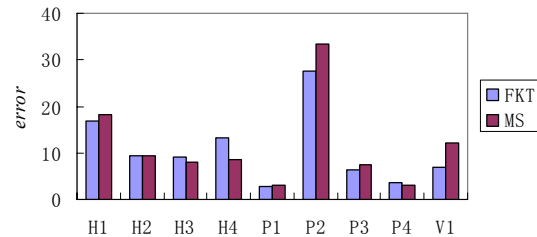
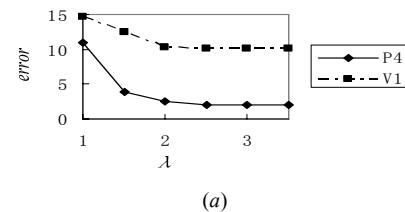
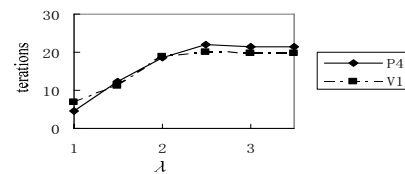


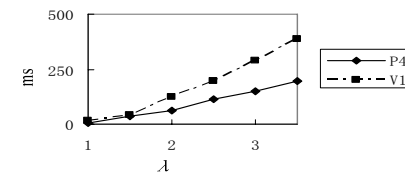
Figure 11. Average Tracking Error in MS and FKT



(a)



(b)



(c)

Figure12. Tracking performance with different  $\lambda$  on video P<sub>4</sub> an V<sub>1</sub>, (a) Average error (pixels), (b) Iteration times, (c) Average time cost (ms)



Figure13 experiment results on 20<sup>th</sup>, 30<sup>th</sup>, 70<sup>th</sup>, 120<sup>th</sup> frame of P4 ( $\lambda=2$ )



Figure14 experiment results on 20<sup>th</sup>,30<sup>th</sup> frame of V1( $\lambda=1$ )

**E. Time Complexity**

Average time cost of classic mean shift is  $N\max(nc_s, m)$  where  $\max(x, y)$  means the maximum value between  $x$  and  $y$ .  $N$  indicate the average iteration times and  $n$  is the number of pixel in object area.  $c_s$  is the extra cost for division and root. Supposed value of  $m$  fall in a similar range with number of pixel in object area, the average cost is  $Nn_h c_s$ [2]. Compared to classic mean shift, in our algorithm,  $\mu_u$  is needed to be computed in each iteration and its cost is  $\max(n', m)$  where  $n'$  is the number of pixel in outer rectangular area. So the time cost of presented algorithm is  $N\max(n', nc_s, m)$ . if  $n' \leq nc_s$ , our algorithm will not increase the time complexity but will increase some computing time.

**VII. CONCLUSION**

This paper mainly focus on the problem that background pixels in object model have an effect on the tracking precision. For reducing the localization error of object tracking, a fuzzy kernel representation is presented and a strategy for fuzzy membership function is given, moreover, this strategy are discussed by experiments. Although our method can improve the tracking precision, there are other factors having an effect on localization, such as Taylor approximate expansion formula and color histogram which lacks spatial information. These are our future work.

**ACKNOWLEDGMENT**

This work was supported by NSFC in China under Grant No. 61170102, the Natural Science Fund of Hunan province in China under Grant No.11JJ3070, No.10JJ3002 and No.11JJ4050, and the Science & Technology Foundation of Hunan Province in China under Grant No.2011FJ3184.

**REFERENCES**

[1] D. Comaniciu and P. Meer, "Mean shift analysis and applications", Proc. of Int'l Conf. on Computer Vision, vol.2, pp.1197-1203, September 1999

[2] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking", IEEE Trans. on PAMI, vol.25, No.5, pp.564-577, 2003.

[3] N. Peng, J. Yang, Z. Liu, et al, "Automatic Selection of Kernel-Bandwidth for Mean-Shift Object Tracking", Journal of Software, vol.16, No.9, pp.1542-1550,2005. (in Chinese)

[4] R.T. Collins, Y. Liu and M. Leordeanu, Online Selection of Discriminative Tracking Features. IEEE Trans. on PAMI, vol.27, No.10, pp.1631-1643, 2005.

[5] Z. Feng, N. Lu and L. Li, "Research on Image Matching Similarity Criterion Based on Maximum Posterior Probability", ACTA AUTOMATICA SINICA, vol.33, No.1, pp.1-8, 2007. (in Chinese).

[6] Z. Zhu, H. Lu and Z. Li, "Novel Object Recognition Based on Hyp-othesis Generation and Verification". Proc. of the 3rd Int. Conf. on Image and Graphics, pp.88-91, December. 2004.

[7] Z. Zivkovic and B. Krose, "An EM-like algorithm for color-histogram-based object tracking", Proc. of IEEE Comp. Society Conf. on Computer Vision and Pattern Recognition, vol.1, pp.798-803, June 2004.

[8] D. Xu, Y. Wang, J. An, "Applying a New Spatial Color Histogram in Mean-Shift Based Tracking Algorithm". Proc. of Image and Vision Computing New Zealand, November 2005.

[9] A. Chan and N. Vasconcelos, Modeling, "Clustering, and Segmenting Video with Mixtures of Dynamic Textures", IEEE Trans. on PAMI, vol.30, No.5, pp.909- 926, 2008.

[10] J. Han and K. Ma, "Fuzzy Color Histogram and Its Use in Color Image Retrieval", IEEE Trans. on Image Processing, vol.11, No.8, pp.944-952, 2002.

[11] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis". IEEE Trans. on PAMI, vol.24, No.5, pp.603-619, 2002.



**Zhiqiang Wen** Hunan Province, China. Birthdate: Sep., 1973. is technology of computer application Ph.D., graduated from the Central South University. And research interests on computer vision, digital image processing, and robotics.

He is an associate professor of School of Computer and Communication, Hunan University of Technology.



**Yongxin Long** Hunan Province, China. Birthdate: Jan., 1966. is Communication and information systems Ph.D., graduated from Zhejiang University. And research interests on computer vision, digital signal processing, automatic control, and robotics.

He is an associate professor of School of Computer and Communication, Hunan University of Technology.



**Zhaoyi Peng** Hunan Province, China. Birthdate: July, 1968. is Software Engineering M.S, graduated from the Central South University. And research interests on digital image processing and pattern recognition.

He is an associate professor of School of Computer and Communication, Hunan University of Technology