Research Journal of Applied Sciences, Engineering and Technology 5(11): 3212-3218, 2013
DOI:10.19026/rjaset.5.4559
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: October 17, 2012 Accepted: December 10, 2012 Published: April 05, 2013

Research Article

Robust Visual Tracking via Fuzzy Kernel Representation

Zhiqiang Wen, Yongxin Long and Zhaoyi Peng
[1]School of Computer and Communication, Hunan University of Technology, Zhuzhou, Hunan, 412008, China

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.

Keywords: Fuzzy factor, fuzzy kernel histogram, fuzzy membership function, visual tracking

INTRODUCTION

Since mean shift and its application appeared in 1999 (Comaniciuand Meer, 1999), it has been widely used for object tracking, image segmentation, pattern recognition, clustering, filtering, etc. Comaniciu *et al*. (2003) 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*. (2005) 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*. (2005) 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*. (2007) 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*. (2004) used the geometry global shape context information to improve the description of object. Zivkovic and Krose (2004) used the color histogram of 5-degrees of freedom to track the revolving object. Other methods are as spatial color histogram (Xu *et al*., 2005), mixed model of dynamic texture (Chan and Vasconcelos, 2008), etc.

In this study, 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 (Han andMa, 2002) 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.

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.

Corresponding Author: Zhiqiang Wen, School of Computer and Communication, Hunan University of Technology, Zhuzhou, Hunan, 412008, China

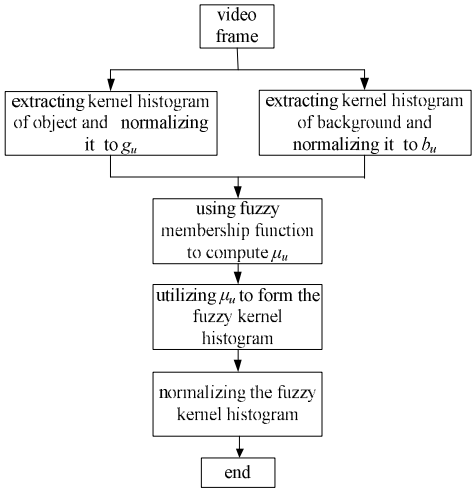

Fig. 1: Process of creating fuzzy kernel histogram

Definition 1: If A denote the set of object pixel and universe U represent the color features in object model, a fuzzy set A on universeU is defined as A = {μA (x), x∈U}, where μA(x)∈[0,1] named as membership of set A on U.

In above definition, μA (x) = 0 (μA (x) = 1) means membership is not attributed (completely attributed) to A on U and μA (x) = 0.5 represents the fuzzy boundary point of set A on U. According to definition 1, μ(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:

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

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

In (1) and (2), both q = {qu} and p(y) = {pu (y)} u = 1,…, m are object model and object candidate respectively where y is the center location of object candidate, satisfying ∑qu = 1 and ∑pu = 1. In addition, m is the number of bins and δ (•) is the Kronecker delta function. {x*i}I = 1,…, n is the normalized pixel locations in object region which is centered at 0. Function b: R2→ {1,…, m} associates to the pixel at location x*i the index b (x*i) of its bin in the quantized feature space. Both C and CH are the normalization constant. Both n and nH are the total pixel number in object region. k (·) is a kernel function and H is the bandwidth matrix. For (1) and (2), the key problem is how to acquire μu. Membership function of fuzzy set on universe U is actually the real function that x∈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 U is in real number field, for fuzzy set A in R, fuzzy membership function μA (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 μu.

Fig. 2: Center-surround approach

FUZZY MEMBERSHIP FUNCTION

In kernel tracking, according to the definition of fuzzy set, membership factor μu should show the degree that kernel histogram features attribute to the object, so μu = 0 means the u[th] background feature is similar to corresponding object feature which should be omitted. μu = 1 shows the uth 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 μu should be computed via both object model and background pixel. Basic process of creating fuzzy kernel histogram in this study is shown in Fig. 1. In Fig. 1, after kernel histogram of object color and the histogram of background color are extracted respectively from the video frame named gu and bu, a fuzzy membership function is used to form fuzzy membership factor μu by utilizing both gu and bu. Then, the fuzzy kernel histogram is created by both (1) and (2). In this study we use the center-surround approach (Collins *et al*., 2005), 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 Fig. 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×wpixels and the outer rectangular is rh×rw size, where r>1. In this study 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 gu and bu, where 1≤u≤m. After acquiring gu and bu,

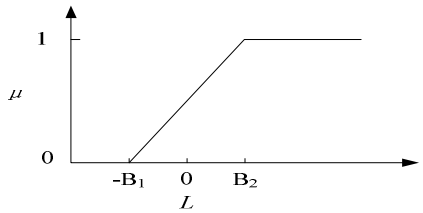

Fig. 3: L-μ curve

the next step is to decide the fuzzy membership function according to both gu and bu. A ratio strategy will be given to create the fuzzy membership functions in next paragraph.

We present a fuzzy membership function as shown in (3), which is piecewise function of linear transformation. In (3), both B1 and B2 are positive constants limiting distribution of μu in [0, 1] and L (u) is a new feature. The L-μ curve about (3) is shown in Fig. 3. In Fig. 3, μu = 0, if L (u) ≤-B1 while μu = 1 if L(u)≥B2. For both object and background feature, a mapping function f(gu, bu) is used to form the new feature L(u) for better discrimination between object and background, namely L (u) = f (gu, bu). But how to select function f (gu, bu)? It requires that L (u) must describe no background feature, but the object feature in object model. That is to say, large bu describes the background pixels and L (u) is a minimum value. On the other hand, small bu shows gu is the object feature which should be reserved. According to above, f (gu, bu) is decided by the log likelihood ratio of the feature distributions as shown in (4):

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

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

where, δ is a small value (δ = 0.1×10-10 in this study) 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 is 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 occurs 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.

TRACKING BASED ON FUZZY KERNEL HISTOGRAM

In kernel tracking, Bhattacharyya between p (y) and q is ρ (y) ≡ ρ [p (y), q]. After using Taylor expansion on Bhattarchayya 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 (·) = -k'(·). So the Fuzzy Kernel Tracking (FKT) algorithm for finding object position in tth frame can be described as follows:

Step 1: Initialize the object location y0 in the current frame and other parameters H, m, B1, B2, small real number ε, k = 0, maximum iterations N, etc.

Step 2: Compute qu according to (2) in the sample frame.

Step 3: Compute pu (yk+1) according to (1) in next frame, then calculate weight wi.

Step 4: Find the next location y'k+1 of the object candidate according to (5).

Step 5: Correct the location according to the moving information of object, namely yk+1 = y'k+1+Δt·vk, where vk = (y'k+1-yk)/Δk. For the simplicity, let Δk = 1.

Step 6: If ||yk+1-yk||<ε or k<N, stop algorithm, else let k = k+1 and go to Step 3. The output value yk in FKT algorithm is regarded as the object position yt in tth frame. Parameter k describes the iterations of tracking algorithm. Moreover, in Step 1, popularly y0 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.

PARAMETER SELECTION

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

- **Selection of kernel:** Three kernels frequently used in object tracking are Gaussian kernel k (x) = e-x, Epanechnikov kernel:

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

and Truncated Gaussian kernel:

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

There are three reasons for selecting truncated Gaussian kernels.

- 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
- For Epanechnikov kernel, mean shift will turn to:

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

which isn't proved convergent:

- For truncated Gaussian kernel, the computing time will be reduced and its convergence has been proved (Comaniciu and Meer, 2002).

- **Bandwidth matrix:** Using fully parameterized H increases the complexity of the estimation and in practice, H is chosen either as H = diag [h21,…, h2d] or proportional to the identity matrix H = h2I. Let hprev is the bandwidth in previous frame and there are three bandwidth values as h = hprev, h = hprev+Δh and h = hprev-Δh where Δh = 0.01 hprev. A simple method (Comaniciu *et al*., 2003) is to find optimal bandwidth hopt 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 y, width and height. For the simplicity, H = diag [hx, hy], where hx and hy are the width and height of object respectively. After the object center is found by FKT algorithm, optimum hx and hy are searched locally, so that Bhattacharyya coefficient ρ (y) reaches a maximum. The search region cannot be set too large; otherwise, the real-time is bad.
- **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 pu (y0) which means the distance between y0 and 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 study. The method is yt+1 = yt+Δt·vt, where Δt is the time interval between two frames and vt is the movement speed of object between tth and t+1th. Speed vt can be computed by formula vt = (yt+1-yt)/Δt. for simplicity and the equal time interval, let Δt = 1. Furthermore, in (3), let B1 = log (5.0) and B2 = log (2.0), we can get good result.

Fig. 4: Dataset, (from top-left to bottom-right) Head object (H1, H2, H3, H4); Pedestrians (P1, P2, P3, P4); Vehicle (V1)

Fig. 5: Experiment results of FKT and MS on H1 video

EXPERIMENTAL RESULTS

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 (H1, H2, H3 and H4), pedestrians (P1, P2, P3 and P4) and vehicle (V1). H1, H2, H3 and H4come fromthepublicdataset1, P1 s a part of PETS2001 dataset2 and P2 is from CAVIAR dataset3 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 H1, H2, H3 and H4, the velocity of head movement is quick and sometimes changes suddenly. There are shields in H4. In P2, the scene is intricate and daylight illumination may vary. In P1, P3 and P4, the objects are moving slowly and there is little disturbance of daylight illumination. In V1, velocity of the car is quick. In this study, we use VC++6.0 and Open CV as programming tool and assume both the object and its initial location have been known.

- **Performance evaluation:** In order to compare the performance of existing tracking methods, tracking error estimation is used in this study. Tracking error is defined as the deviation between estimating parameters of tracking method and real parameters.

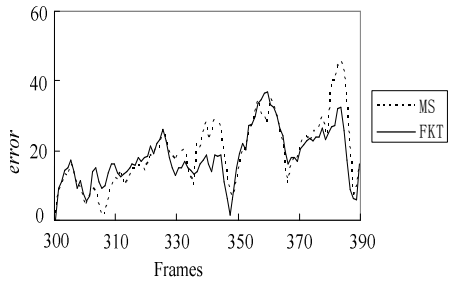

Fig. 6: Error of object location in some frames

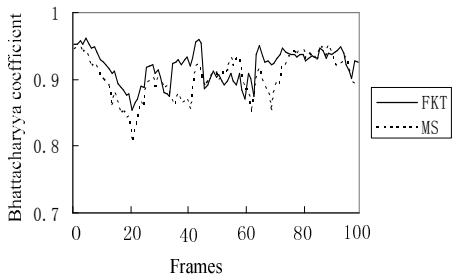

Fig. 7: Comparison of Bhattacharyya coefficient value

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

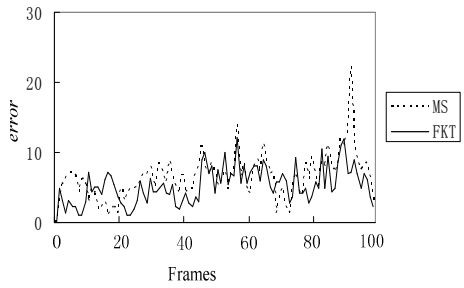

Fig. 9: Error of object location in each frame

We define tracking error of t^{th} frame as: errort = ||xt-yt||2 (pixels), where yt (yit, i = 1, 2) is the parameter vector of object in t^{th} acquired by tracking method and xt (xit, i = 1, 2) is the real parameter vector of object in t^{th}. In our experiments, real parameter vectors of dataset H1, H2, H3, H4 and P1 are come from the data provided by E. Maggio[4], while the real parameter vector of other datasets is obtained via manual method. In addition, we define the average error of errort as error in some sequential frames.

- **Experiment results:** The experiment results of our method (FKT) and Mean Shift (MS) (Comaniciu *et al*., 2003) on H1 video are showed respectively in Fig. 5, 6 and 7. Figure 5 shows the results in 300th, 312th, 330th, 342nd, 375th, 388th 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 Fig. 6 which shows that in the vicinity of 53rd, 94th 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 Fig. 7. From Fig. 11, we know that the average error of FKT is less then that of MS. In the same way, more experiments on P3 are showed in Fig. 8 and 9.

Above figures show the results in 300th, 312th, 330th, 342nd, 375th, 388th frame respectively, where black rectangle denotes MS and white rectangle denotes FKT.

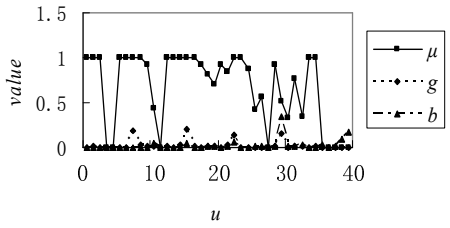

Fig. 10: A demonstration about fuzzy factor

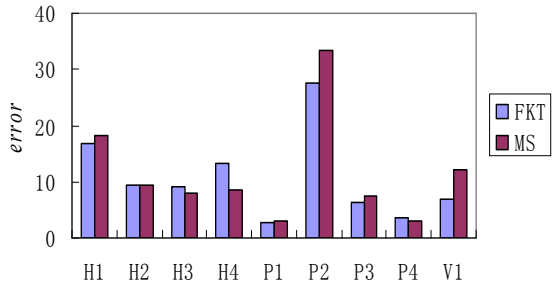

Fig. 11: Average Tracking Error in MS and FKT

Figure 10 shows a demonstration about fuzzy factor μ in the first frame of H1 video. In Fig. 10, μu satisfies 0≤μu≤1. gu>>bu (in experiment, gu>2*bu) means μu = 1. That is to say, gu is the object color and should be kept. bu>>gu (in experiment, bu>5*gu) shows μu = 0. This means that gu 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 H1, H2, P1, P2, P3, V1 and V2, but larger than that of MS on other videos. According to the 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.

- **Effect of parameter λ:** Parameter λ will have an impact on performance as positioning accuracy, iteration times and runtime. So in this section, we will discuss how to select parameter λ by experiments. In our experiments, the tracking results is achieved by our algorithm respectively for λ ∈ [1, 4] as shown in Fig. 12. Fig. 12a shows the average error with different λ. Figure 12b and c show the iteration times and average time (Intel® core™ 2 DUO CPU 2.19GHZ and memory 2.98GB) respectively with λ∈ [1, 4]. From Fig. 12, we can conclude that small λ will result in bad position accuracy, little iteration times and quick computation speed but large λ will lead to bad real-time performance. Furthermore, λ is important for the performance of object tracking against occlusion. If λ is too small, the object is easily lost. If λ is too large, the computed time is too long and real-time performance is bad. Generally λ = 1.5-2.5. λ = 2 in our experiment. Figure 13 and 14 show the tracking result of FKT when λ = 2 (λ = 1) on video P4 from which we can conclude that λ = 2 can provide a good ability against occlusion but λ = 1 will lead to object missed.

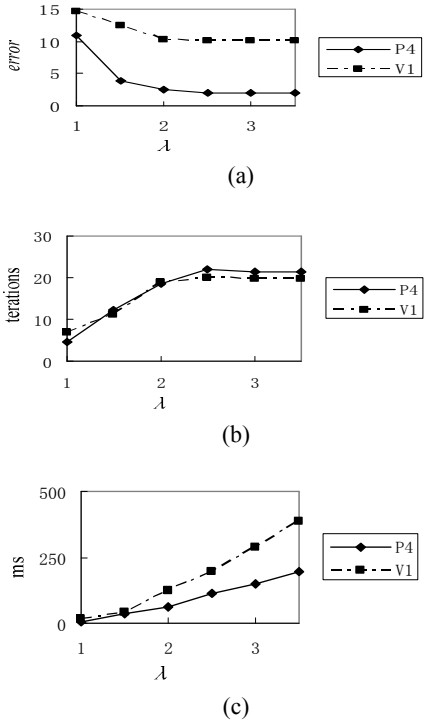

Fig. 12: Tracking performance with different λ on video P4 a V1, (a) Average error (pixels), (b) Iteration times, (c) Average time cost (ms)

Fig. 13: Experiment results on 20th, 30th, 70th, 120th frame of P4 (λ =2)

Fig. 14: Experiment results on 20th, 30th frame of V1 (λ = 1)

- **Time complexity:** Average time cost of classic mean shift is Nmax (ncs, 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. cs 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 Nnhcs (Comaniciu *et al*., 2003). Compared to classic mean shift, in our algorithm, μu 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 Nmax (n', ncs and m). If n'≤ncs, our algorithm will not increase the time complexity but will increase some computing time.

CONCLUSION

This study 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 study.

ACKNOWLEDGMENT

This study 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 and Technology Foundation of Hunan Province in China under Grant No.2011FJ3184.

REFERENCES

Chan, A. and N. Vasconcelos, 2008. Modeling, clustering and segmenting video with mixtures of dynamic textures. IEEE T. PAMI, 30(5): 909-926.

Collins, R.T., Y. Liu and M. Leordeanu, 2005. Online selection of discriminative tracking features. IEEE T. PAMI, 27(10): 1631-1643.

Comaniciu, D. and P. Meer, 1999. Mean shift analysis and applications. Proceeding of the International Conference on Computer Vision, 2: 1197-1203.

Comaniciu, D. and P. Meer, 2002. Mean shift: A robust approach toward feature space analysis. IEEE T. PAMI, 24(5): 603-619.

Comaniciu, D., V. Ramesh and P. Meer, 2003. Kernel-based object tracking. IEEE T. PAMI, 25(5): 564-577.

Feng, Z., N. Lu and L. Li, 2007. Research on image matching similarity criterion based on maximum posterior probability. Act. Automat.Sin., 33(1): 1-8. (In Chinese)

Han, J. and K. Ma, 2002. Fuzzy color histogram and its use in color image retrieval. IEEE T. Image Process., 11(8): 944-952.

Peng, N., J. Yang and Z. Liu, 2005. Automatic selection of kernel-bandwidth for mean-shift object tracking. J. Softw., 16(9): 1542-1550. (In Chinese)

Xu, D., Y. Wang and J.An, 2005. Applying a new spatial color histogram in mean-shift based tracking algorithm. Proceeding of the Image and Vision Computing. New Zealand.

Zhu, Z., H. Lu and Z. Li, 2004. Novel object recognition based on hypothesis generation and verification. Proceeding of the 3rd International Conference on Image and Graphics, pp: 88-91.

Zivkovic, Z. and B. Krose, 2004.An EM-like algorithm for color histogram based object tracking. Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1: 798-803.