Australian Journal of Basic and Applied Sciences, 5(3): 592-600, 2011
ISSN 1991-8178

An Efficient Tracking Algorithm Based on Spatial Kernel and FCM Classifier

[1]Mohammad Yosefi, [2]Mehran Erza, [3]Nima Aliakbari and [4]Saeid Khoshguian

[1,3,4]Department of Electrical and Computer Engineering, Babol Noshiravani University of Technology P.O. 484, Babol, Iran
[2]Department of Micro and Nano Electronic, University Of Joseph Fourier
P.0. 38000, Grenoble, France

Abstract: A modified movable object tracking algorithm which uses the flexible Metric Distance Transform kernel and FCM Classifier is proposed and tested. The target shape which defines the d_n Distance Transform is found based on conventional statistical parameters as feature vector extraction and Fuzzy C-Mean (FCM) classifier to differentiate tracked target from background. This replaces the more usual Epanechnikov kernel (E-kernel), improving target representation and localization without increasing the processing time, minimizing the similarity measure using the Bhattacharya coefficient. The algorithm is tested on several image sequences and shown to achieve robust and reliable frame-rate tracking.

Key words: Movable Object tracking, Metric Distance Transform, Mean Shift, Fuzzy C-Mean (FCM), Bhattacharyya coefficient, Epanechnikov kernel.

INTRODUCTION

Real-time object tracking is the critical task in many Computer vision applications such as surveillance Greiffenhagen *et al.,*, (2001) perceptual user interfaces Bradski, (1998), augmented reality Ferrari *et al.,* (2001), smart rooms Intille *et al.,* (1997), object-based video compression Bue *et al.,* (2002) and driver assistance Handmann *et al.,* (1998). Two major components can be distinguished in a typical Visual tracker. Target Representation and Localization is mostly a bottom-up process which has also to cope with the changes in the appearance of the target. Filtering and Data Association is mostly a top-down process dealing with the dynamics of the tracked object, learning of scene priors, and evaluation of different hypotheses. The way the two components are combined and weighted is application dependent and plays a decisive role in the robustness and efficiency of the tracker. For example, face tracking in a crowded scene relies more on target representation than on target dynamics, while in aerial video surveillance, e.g., Wildes *et al*, (2001) the target motion and the ego-motion of the camera are the more important components. Therefore, it is desirable to ensure that the tracker is as efficient as possible. Kernel-based density estimation techniques for Computer vision have attracted a great deal of attention. One example is the mean shift (MS) technique which has been applied to image segmentation, visual tracking, etc. (Greiffenhagen *et al.,*, 2001; Bradski, 1998; Ferrari *et al.,* 2001; Intille *et al.,* 1997; Bue *et al.,* 2002; Handmann *et al.,* 1998; Wildes *et al,* 2001). MS is a versatile nonparametric density analysis tool introduced in (Jamasbi, *et al.,* 2007; Comaniciu and Meer, 2002; Comaniciu *et al.,* 2003). In essence, it is an iterative mode detection algorithm in the density distribution space. The MS algorithm moves to a kernel-weighted average of the observations within a smoothing window. This computation is repeated until convergence is attained at a local density mode. This way the density modes can be elegantly located without explicitly estimating the density. Cheng Comaniciu and Meer, (2003) notes that MS is fundamentally a gradient ascent algorithm with an adaptive step size. Fashing and Tomasi, (2003) show the connection between MS and the Newton–Raphson algorithm. They also discover that MS is actually a quadratic bound optimization both for stationary and evolving sample sets. MS is also a fixed-point iteration procedure. Since Comaniciu et al. Bradski, (1998) first introduced MS-based object tracking, it has proven to be a promising alternative to popular particle altering based trackers (Cheng, 1995; Bhattacharyya, 1943). A number of improvements to the method have been reported in the literature. In Ferrari *et al.,* (2001), the selection of kernel scale via linear search is discussed. Intille *et al.,* (1997) reformulates the tracking framework as a gen- eral form of joint feature-spatial distributions Wildes *et al.,* (2001). Compared with the approach of Comaniciu et al., the advantage is that spatial structure information of the tracked region is incorporated. In

Corresponding Author: Mohammad Yosefi, Department of Electrical and Computer Engineering, Babol Noshiravani University of Technology P.O. 484, Babol, Iran

Bue, *et al.,* (2002), multiple spatially distributed kernels are adopted to accurately capture changes in the target's orientation and scale. Another approach is developed in Stauffer and Grimson, (1999) for the same purpose. Furthermore, Swain and Ballard, (1991) present a theoretical analysis of the similarity measure and arrive at a criterion, leading to kernel design strategies with prevention of singularity in kernel visual tracking. All the above mentioned trackers adopt MS or similar optimization strategies. Despite successful applications, MS trackers require that the displacement of the tracked target in consecutive frames to be small because the search is initialized by the detected location of the target in the previous frame. Larger inter- frame displacements will lead the tracker to become trapped in spurious locations in the multimodal density distribution space because MS is a local optimization method. So in many situations, however, we seek the global mode of a density function. The standard MS tracker assumes that the initialization point falls within the basin of attraction of the desired mode. On the other hand, in basic MS algorithm, the feature histogram-based target representations are regularized by spatial masking with a descent isotropic kernel. In most instances, the target does not have radial symmetry, so the use of a E-kernel includes foreground as background, or background as foreground pixels, or both. So non-included target pixels are weighted. Hence, the local and global modes may not correspond and the tracker is likely to fail. The main contribution of the paper is to introduce a new framework for efficient tracking of rigid objects. We show that how such a kernel-based object tracking algorithm can be improved by using a kernel based on the d_a Distance Transform. Statistical mean and standard deviation are used to obtain feature vector on the target in first frame to use Fuzzy C-Mean (FCM). The central of target localization in each frame is estimated by MS and then FCM is applied to differentiate the target from background. Weighting of segmented target pixels is done by metric Distance Transform which is used to estimate of target localization for next frame. We present experiments that demonstrate the superior performance of this approach in comparison with the basic MS algorithm or self-FCM classifier by measuring accuracy, robustness and stability. The remainder of the paper is organized as follows. Section 2 explains the kernel density estimating and means shift analysis. Fuzzy C-Mean (FCM) classifier is explained in Section 3. Section 4 introduces metric distance transforms. Section 5 explains the experimental settings and the experimental results will be reported. Finally, a conclusion and discussion are presented in Section 6.

Kernel Density Estimation:

Motion of the target in two sequential single images describe with good approximation using mean shift algorithm Comaniciu *et al.,* (2003) At the first stage, we make a model of the target using weighing Histogram, which exploits both color and location data. The region of target is supposed as elliptic in the first frame then points inside of elliptic are normalized to unit circle with 0 center. Marginal pixels are less important than central pixels because they are influenced by occlusion and interference with background. In order to consider this importance, we allocate a weight to each pixel based upon distance of that pixel from center. Therefore weighing Histogram is obtained as below:

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

Where *b* function relates corresponding index to each pixel. *K* is a convex, uniformly decreasing and isotropic function. *C* constant is defined in order to provide $\sum_{u=1}^{m}\hat{q}_u = 1$ .We search an elliptical region in new frame on which is most similar to targeted one. We search our target in new frame around the target location in previous frame and estimate most similarities as new location of the target because target motion in two sequential frames is not considerable. This exhaustive search is time consuming, so utilizing of proper recursive algorithm is recommended. We should consider another parameter other than target location. This parameter is object scale (*h*) which modifies elliptic size in each frame. We define similarity criteria for comparing two Histograms. In this paper we use Bhattacharya distance which defined below:

$$\hat{\rho}(y) \equiv \rho\left[\hat{p}(y),\hat{q}\right] = \sum_{u=1}^{m}\sqrt{\hat{p}_u(y)\hat{q}_u} \tag{2}$$

If in previous frame target location $\hat{y}_0$ and target scale *h* are, in current frame similarity of Histogram around $\hat{y}_0$ will be calculated through Taylor expansion:

$$\rho\left[\hat{p}(y),\hat{q}\right]\approx\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u\left(\hat{y}_0\right)\hat{q}_u}+\frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(y)\sqrt{\frac{\hat{q}_u}{\hat{p}_u\left(\hat{y}_0\right)}} \tag{3}$$

Let $\left\{x_i\right\}_{i=1\ldots n_h}$ be the normalized pixel locations in the current frame. Using the same kernel profile *k(x)*, but with bandwidth h, the probability of the future *u=1…m* in the target candidate is given by

$$\hat{p}_u(y)=C_h\sum_{i=1}^{n_h}k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)\delta[b(x_i)-u] \tag{4}$$

Recalling (4) results in

$$\rho\left[\hat{p}(y),\hat{q}\right]\approx\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u\left(\hat{y}_0\right)\hat{q}_u}+\frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(y)\sqrt{\frac{\hat{q}_u}{\hat{p}_u\left(\hat{y}_0\right)}} \tag{5}$$

Where

$$w_i=\sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u\left(\hat{y}_0\right)}}\,\delta\left[b\left(x_i\right)-u\right] \tag{6}$$

We should maximize second term in (5) in order to have maximum similarity.

We consider this term as Kernel estimation *k(x)* of a probability density function and also use mean shift algorithm for calculating maximum of this equation. In this algorithm current location moved to the new one as below:

$$\hat{y}_1=\frac{\sum_{i=1}^{n_h}x_i w_i g\left(\left\|\frac{\hat{y}_0-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h}w_i g\left(\left\|\frac{\hat{y}_0-x_i}{h}\right\|^2\right)} \tag{7}$$

***Background-Weighted Histogram*:**

The background information is important for at least two reasons. First, if some of the target features are also present in the background; their relevance for the localization of the target is diminished. Second, in many applications, it is difficult to exactly delineate the target, and its model might contain background features as well. At the same time, the improper use of the background information may affect the scale selection algorithm, making impossible to measure similarity across scales, hence, to determine the appropriate target scale. A good approach is to derive a simple representation of the background features and to use it for selecting only the salient parts from the representations of the target model and target candidates Comaniciu *et al.,* (2003)

Let $\{\hat{o}_u\}_{u=1\ldots m}$ (with $\sum_n^m \hat{o}_u = 1$) be the discrete representation (histogram) of the background in the feature space and $\hat{o}_v$ be its smallest nonzero entry. This representation is computed in a region around the target. The extent of the region is application dependent and we used an area equal to three times the target area. The weights:

$$\left\{ v_u = \min\left(\frac{o_v}{\hat{o}_u}, 1 \right) \right\}_{u=1...m} \tag{8}$$

are similar in concept to the ratio histogram computed for back projection Swain and Ballard, (1991). However, in our case, these weights are only employed to define a transformation for the representations of the target model and candidates. The transformation diminishes the importance of those features which have low v_u, i.e., are prominent in the background. The new target model representation is then defined by:

$$\hat{q}_u = CV_u \sum_{i=1}^{n} k\left(\left\| x_i^* \right\|^2 \right) \delta\left[b\left(x_i^* \right) - u \right] \tag{9}$$

Compare with (2) similarly, the new target candidate representation is:

$$\hat{p}_u(y) = C_h V_u \sum_{i=1}^{n_h} k\left(\left\| \frac{y - x_i}{h} \right\|^2 \right) \delta[b(x_i) - u] \tag{10}$$

The target localization algorithm is described as follow:

1- Calculate the target model $\left\{ \hat{q}_u \right\}_{u=1...m}$ and its position $\hat{y}_0$ in first frame.

2- Calculate $\left\{ \hat{p}_u\left(\hat{y}_0 \right) \right\}_{u=1...m}$ and compute $\rho\left[\hat{p}_u\left(\hat{y}_0 \right), \hat{q}_u \right]$ in current frame.

3- Obtain the weight w_i using (6).

4- Finding the new target location using (7).

5- If $\left\| \hat{y}_1 - \hat{y}_0 \right\| \langle \varepsilon$ Stop otherwise $\hat{y}_0 \leftarrow \hat{y}_1$ and go to step1.

The fuzzy c-mean algorithm:

The fuzzy c-means (FCM) algorithm was introduced by J.C. Bezdek, (1981)]. The idea of FCM is using the weights that minimize the total weighted mean-square error:

$$J_m(U,V) = \sum_{k=1}^{n} \sum_{j=1}^{l} u_{ij}^m \| x_k - V_j \|^2 \tag{11}$$

Where *Vj* is mean of j' class and renewed by:

$$V_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m} \tag{12}$$

The FCM allows each feature vector to belong to every cluster with a fuzzy truth value (between 0 and 1), which is computed using Equation (13). The algorithm assigns a feature vector to a cluster according to the maximum weight of the feature vector over all clusters.

$$u_{ik} = \frac{1}{\sum_{j=1}^{l} \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}} \tag{13}$$

Repeat Steps 3)–5) till termination. The termination criterion is as follows:

$$\|J_{new} - J_{old}\| < \varepsilon \tag{14}$$

Where $\|\cdot\|$ is the Euclidean norm, J is a mean square error, and is a small number that can be set by the user.

Kernel Based on the Metric Distance Transform:

The equivalence of the mean shift procedure to gradient ascent on the similarity function holds for kernels that are radially symmetric, non-negative, non-increasing and piecewise continuous over the profile S.rastegar *et al.*, (2009). A radially symmetric kernel can be described by a 1D profile rather than a 2D (or higher order) image. The usual choice for K is the optimal Epanechnikov kernel (E-kernel) Bhattacharyya, (1943) that has a uniform derivative of G=1 which is also computationally simple. However, in tracking an object through a video sequence and applying the mean shift algorithm to move the position of the target window, the bounds of the domain R2 are altered on each successive application of the algorithm. In most instances, the target does not have radial symmetry, so the use of an E-kernel includes foreground as background, or background as foreground pixels, or both. Depending on the shifting of pixels between background and foreground, and on the similarity of the two color distributions (in a worst case the background has similar properties to the target), then multiple modes are formed in the pdf. In this case local and global modes don't correspond and the tracker is likely to fall. Therefore, our contribution is to use a distance transform (DT), matched to the shape of the tracked object, as a kernel function. Although this kernel does not change shape through the sequence, it can change size, scaling as the subject expands or contracts in the camera field of view. For the DT each foreground pixel is given a value that is a measure of the distance to the nearest edge pixel. The edge and background pixels are set to zero. We use the normalized Metric d_a distance Transform (MDDT) rather than the true Euclidean distance and isotropic descent kernel, as it is an efficient approximation. The MDDT kernel better represents the color distribution of the tracked target, yet retains the more reliable centre weighting of the radially symmetric kernels. Let P be a binary picture (defined on grid G) in which

$$< P > = \{p:\ p\epsilon G\ \ \&\, P(p) = 1\} = P^{-1}(1)$$

$$< \bar{P} > = \{p:\ p\epsilon G\ \ \&\, P(p) = 0\} = P^{-1}(0) \tag{15}$$

Are proper subsets of G. For any grid metric d_a, distance transform of P associates with every pixel p of $<P>$ the d_a distance from p to $<P>$We assume that pixels of the background component (i.e., containing all pixels outside of the rectangular region G) all have value 0. The d_4 or d_8 distance transform of p (and others – see later) can be computed by performing a series of local operations while Scanning Gtwice. For any $p\in G$ let $B(p)$ ("before") be the set of pixels (4-or 8-) adjacent to p that precedes p when G is scanned in standard order:

If p has coordinates (x,y), B contains $(x, y + 1)$ and $(x - 1, y)$, and if we use 8-adjacency it also contains $(x - 1, y + 1)$ and $(x + 1, y + 1)$. For first scan we have

$$f_1(p) = \begin{cases} 0 & if\ p\epsilon\langle\bar{P}\rangle \\ \min\ \ f_1(q) + 1 : q\epsilon B(p) & if\ p\epsilon\langle P\rangle \end{cases} \tag{16}$$

Compute $f_1(p)$ for all $p\in G$ in a single standard scan of G; for each p, f1 has already been computed for all of the qs in $B(p)$ (If p is on the top row or in the left column of G, some of these qs are outside G with f^1= 0.)

And for second scan we have:

$$f_2(p) = \min\{\, f_1(p),\ f_2(p) + 1\ : q\epsilon\, A(p)\} \tag{17}$$

Compute $f2(p)$ for all $p\in G$ in a single reverse standard (i.e., right-to-left, bottom-to-top) scan of G (note: each p, f^2 has already been computed for all of the qs in $A(p)$ or is known because they are outside of G).

In this work, d_8 transform is applied to the target which separated from the background by FCM classifier. This weighting can increase the accuracy and robustness of representation of the pdf's as the target moves, excluding the peripheral pixels that occur within a radially symmetric window. Applying the MDDT kernel to the region of interest, and weighting the colour distributions accordingly, we determine whether the exclusion of the erroneous background pixels, for example, from the density Estimate of the target, and giving increased weighting to those more reliable pixels towards the centre, will outweigh the possibility of forming false modes. Of course, although the MDDT may produce false modes, this also occurs with radially symmetric kernels due to badly defined densities. As the scale of the target may change, the size of the Kernel is adapted accordingly. When the exact location of the target is found, we measure the similarity criterion for three values $h = [.9 \;\; 1 \;\; 1.1] \times h_{prev}$ and its maximum is named h_{opt}.h_{new} (Scale for current frame) is computed as bellow:

$$h_{new} = \lambda\, h_{opt} + (1 - \lambda)\, h_{prev} \tag{18}$$

The best result, h_{new} yielding the largest Bhattacharyya coefficient is retained.

In applying the MDDT kernel to the mean shift procedure, due to the object shape changes in the long or short term, we have two options. First we can define the MDDT on the basis of the first frame, and use this for the whole sequence. Second we can segment the subsequent frame and apply a different kernel weighting. The algorithm described above applies to the first option, which is used for the experimental results in the next section. To modify or update for each model frame, for example, the segmentation and MDDT computation code is included inside the outer repeat-until loop.

Adaptive kernels have also been used by Porikli and Tuzel, (2003). Like our approach, their algorithm does not maintain fully the mean-shift convergence conditions Comaniciu and Meer, (2002). However, the MDDT presented here satisfies it partly with its decreasing Profile. Practical tests show that even if theoretical convergence conditions are not fully satisfied, convergence is achieved.

Experimental Results:

In this section, we present the evaluation of the modified mean shift object tracking using the MDDT-kernel in comparison with the radially symmetric E-kernel. We track moving objects, a static object with a moving camera and a combination of the two. All the tests were carried out on a Pentium 4 CPU 3.80 GHz with 2GB RAM. So that it would be reasonable to assume a considerable increase in processing speed if re-implemented with more advanced sets. In the first experiment, we compare the tracking of a moving car in a video sequence that includes 368 frames of 480´ 640 pixels, comparing the normal E-kernel with the MDDT

Kernel. Fig.1shows the value of iteration computed for each frame. In Fig.2 distance function which calculated by the Bhattacharyya coefficient (Eq. 2) is presented. The peak in the E-kernel data is 0.636 which increases to the 0.7 in MDDT. Fig.3 (a) and (b) show some examples, frames 1, 35, 60 and 75, from the whole sequence. In frame 60 some of the original car is still contained within the window, but after the 75nd frame, the car is lost completely in Fig.3 (a), as the tracker finally latches on to another crossing car. This demonstrates that the inclusion of the background of the tracked car (in this case another car) includes pixels that are similar in color space, so that the algorithm fails to identify the correct distribution in succeeding frames and hence follows the wrong target. Fig.4 shows the similarity surfaces made by candidate models in frame 54 with E-kernel and MDDT kernel, respectively. Initial point is center of target model in frame of 50 and extend of simulation is 60×60.

In terms of complexity, computed from 20 executions of the program, the average selected frames per second of the MDDT kernel and the E-kernel are 12.33 and 17.15 respectively, while This time length to 23.42 when FCM is used for target tracking. The maximum numbers of iterations within a single frame are 13 and 20, respectively. The average times per frame are roughly comparable because although the speed of convergence is quicker with the MDDT-kernel, additional processing is required to segment the target window, in order to get more robust and accurate tracking. To further test the robustness of the MDDT-kernel algorithm and convergence properties, the contrast of input selected frames is increased which can result by different environment illumination. From Table 1, which shows quantitative results, the MDDT kernel algorithm needs on average only 7.4 iterations to converge to the optimal result, but the E-kernel needs 13 iterations on average. Again, the greater complexity of computing the MDDT kernel is balanced by the greatly reduced number of iterations, so the processing speed per frame is comparable. As it shown Self-FCM classifier among three methods has most consuming time.

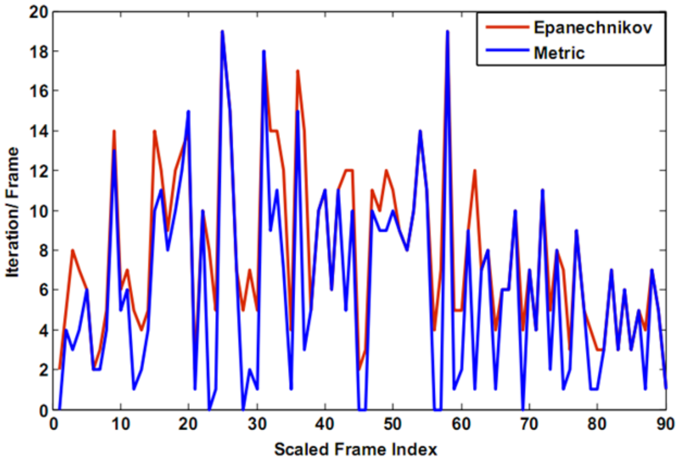


Fig. 1: Iteration Value for selected Frames

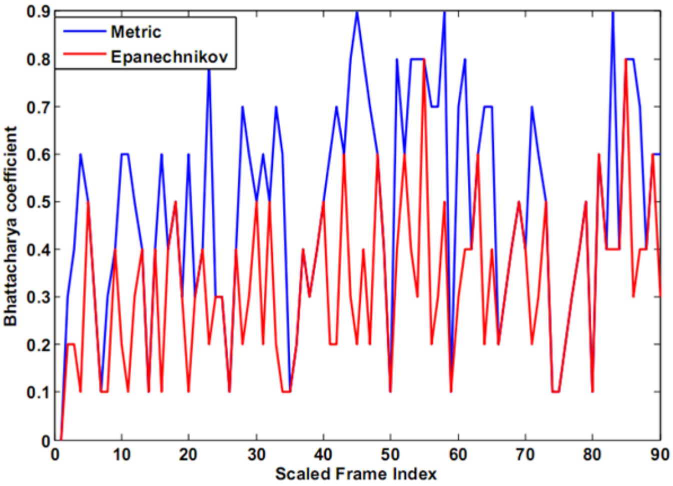
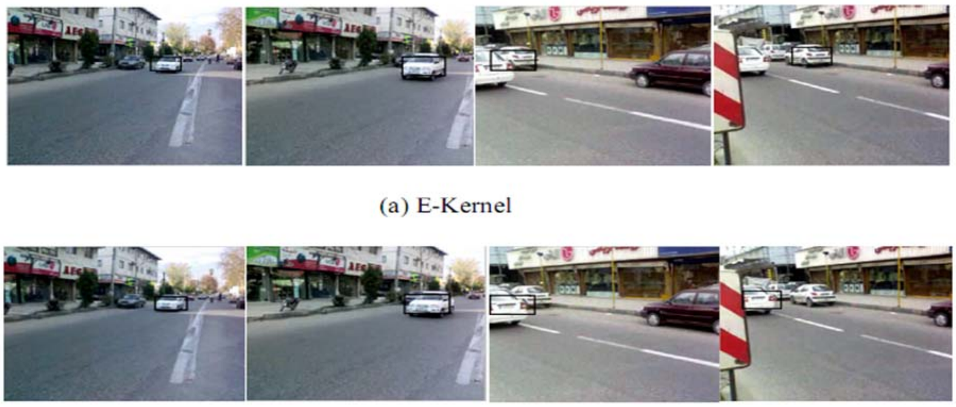


Fig. 2: The Bhattacharya distance values, for the crossing car



(a) E-Kernel

(b) MDDT-Kernel

Fig. 3: Tracking the crossing car

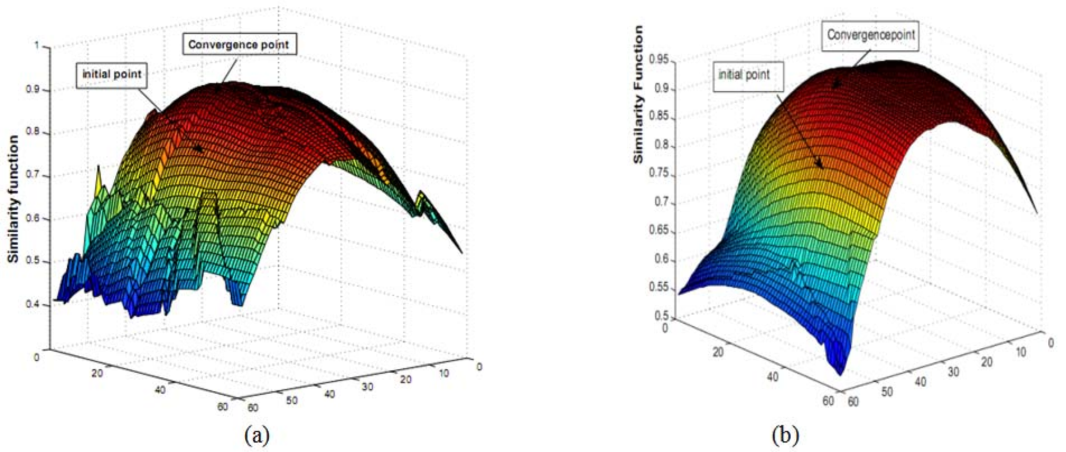


Fig. 4: The similarity surfaces (values of the Bhattacharyya coefficient) for frame 54. The initial points and convergence points are shown. (a) The result from the E-kernel. (b) The result from the MDDT-kernel.

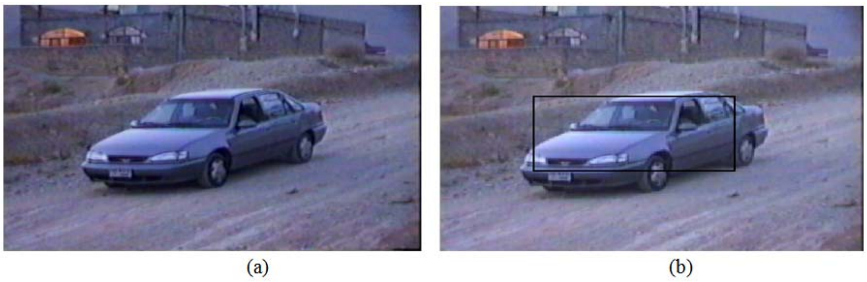


Fig. 5: (a) Original image. (b) Intensity of pixels in selected frame is increased which can result by different environment illumination.

Table 1: Comparison results of consuming time in three methods

Method	Average iterations	CPU time (sec./frame)		
		max	min	mean
E-kernel	13	0.4405	0.3006	0.3705
FCM	-	0.7654	0.4819	0.6236
Proposed Method	7.4	0.3290	0.2404	0.2847

Conclusions:

We have described the implementation of a scaling, normalized Metric d_8 distance kernel as a weighting and constraining function applied to the mean shift tracking algorithm that maximizes the similarity between model and candidate distributions in color space. The tracked target for weighting is differentiated from background by FCM classifier in each frame. In comparison with the E-kernel, used as an exemplar of a radially symmetric function, application of the MDDT-kernel can achieve better results because it can reject false nodes that are caused by the inclusion of changing background pixels. The processing time is sufficiently small for real time operation, as the added cost of foreground-background separation is offset by the more rapid finding of the correct mode. The results presented on a number of video sequences show that the MDDT-kernel algorithm performs well in terms of improved stability, accuracy and robustness on camera motion and partial occlusions.

REFRENCES

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York.

Bhattacharyya, A., 1943. “On a measure of divergence between two statistical populations defined by their probability distributions,” Bulletin of the Calcutta Mathematics Society, 35: 99-110.

Bradski, G.R., 1998. “Computer Vision Face Tracking as a Component of a Perceptual User Interface,” Proc. IEEE Workshop Applications of Computer Vision, pp: 214-219.

Bue, A.D., D. Comaniciu, V. Ramesh and C. Regazzoni, 2002. “Smart Cameras with Real-Time Video Object Generation,” Proc. IEEE Int’l Conf. Image Processing, 3: 429-432.

Carl, G., A. Looney Fuzzy Clustering and Fuzzy Merging Algorithm, CS791q Class Notes, http://www.cs.unr.edu/~looney/.

Carl, G., 1997. Looney Pattern Recognition Using Neural Networks, Oxford University Press, N.Y.

Carl, G., Looney “Chapter 5. Fuzzy Clustering and Merging”, CS791q Class notes, http://www.cs.unr.edu/~looney/.

Cheng, Y.Z., 1995. “Mean shift, mode seeking, and clustering,”IEEE Transactions on PAMI, 17(8): 790-799.

Chen, Z., Z.L husz, I. Wallace, 2007. “video objecting based on chamfer distance transform” Proc. IEEE, 3: 357-360.

Comaniciu, D., and P. Meer, 2002. ‘Mean shift: a robust approach toward feature space analysis’, IEEE Trans. Pattern Anal. Mach. Intell., 24(5): 603-619.

Comaniciu, D., V. Ramesh and P. Meer, 2003. “Kernel-Based Object Tracking,” IEEE Trans. Pattern Analysis and Machine Intelligence, 25(5): 564- 577.

Ferrari, V., T. Tuytelaars and L.V. Gool, 2001. “Real-Time Affine Region Tracking and Coplanar Grouping,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2: 226-233.

Greiffenhagen, M., D. Comaniciu, H. Niemann and V. Ramesh, 2001. “Design, Analysis and Engineering of Video Monitoring Systems:An Approach and a Case Study,” Proc. IEEE, 89(10): 1498-1517.

Handmann, U., T. Kalinke, C. Tzomakas, M. Werner, and W. von Seelen, 1998. “Computer Vision for Driver Assistance Systems,” Proc.SPIE, 3364: 136-147.

Han, M., W. Xu, H. Tao and Y.H. Gong, 2004. An algorithm for multiple object trajectory tracking. In *CVPR,*

Jamasbi, B., A. Motamedi and A. Behrad, 2007. Tracking vehicle targets with large aspect change, IEEE Workshop on Motion and Video Computing.

Intille, S., J. Davis and A. Bobick, 1997. “Real-Time Closed-World Tracking,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp: 697-703.

Maria Colmenares & Olaf WolkenHauer, 2000. “An Introduction into Fuzzy Clustering”, July 1998, last update.

Porikli, F.M., O. Tuzel, 2003. “Human Body Tracking by Adaptive Background Models and Mean-Shift Analysis,” IEEE Int Workshop on Performance Evaluation of Tracking and Surveillance.

Rastegar, S., M, bandarabadi, Y. Toopchi and S. Ghoreishi 2009. ”Kernel Based object tracking using Distance Transform and RVM classifier”-Australian journal of basic and applied sciences- pp: 2778-2790.

Ramze Rezaee, M., B.P.F. Lelieveldt and J.H.C. Reiber, 1998. A New Cluster Validity Index for the Fuzzy c-mean, Pattern Recognition Letters, (Netherlands).

Sabin, M.J., 1987. Convergence and Consistency of Fuzzy c-means /ISODATA Algorithms, IEEE Trans. Pattern Anal. Machine Intell.

Stauffer, C., W.E.L. Grimson, 199. “Adaptive background mixture models for real-time tracking,” Proc. Of IEEE Conf. on Computer Vision and Pattern Recognition, pp: 246-252.

Swain, M. and D. Ballard, 1991. “Color Indexing,” Int’l J. Computer Vision, 7(1): 11-32.

Uri Kroszynski and Jianjun Zhou, 1998. Fuzzy Clustering Principles, Methods and examples, IKS,

Wildes, R., R. Kumar, H. Sawhney, S. Samasekera, S. Hsu, H. Tao,Y. Guo, K. Hanna, A. Pope, D. Hirvonen, M. Hansen and P. Burt, 2001. “Aerial Video Surveillance and Exploitation,” Proc. IEEE, 89(10): 1518-1539.