VIDEO OBJECT TRACKING BASED ON A CHAMFER DISTANCE TRANSFORM

Zezhi Chen[1], Zsolt L Husz[2], Iain Wallace[1], Andrew M Wallace[2]

Joint Research Institute in Signal and Image Processing,
[1]School of Mathematics and Computer Science, [2]School of Engineering and Physical Sciences
Heriot-Watt University, Edinburgh, UK, EH14 4AS

ABSTRACT

This paper describes the use of variable kernels based on the normalized Chamfer distance transform (NCDT) for mean shift, object tracking in colour video sequences. This replaces the more usual Epanechnikov kernel, improving target representation and localization without increasing the processing time, minimising the distance between successive frame RGB distributions using the Bhattacharya coefficient. The target shape which defines the NCDT is found either by regional segmentation or background-difference imaging, dependent on the nature of the video sequence. The improved performance is demonstrated on a number of colour video sequences.

Index Terms— Object tracking, mean shift, Chamfer distance transform, image segmentation, Bhattacharyya coefficient

1. INTRODUCTION

Real-time feature or object tracking in video sequences is an important application in computer vision, often demanding real-time operation. In general, joint spatial and temporal analysis can be used to extract and track regions of interest in the dynamic scene. Some recent exemplars include face tracking in a crowded environment [1], aerial video surveillance [2], human body tracking and behavioural analysis [3, 4]. In the latter example, in particular, only a small fraction of the computational resource can be allocated to tracking, as the rest may be required for high-level tasks such as recognition and understanding of that behaviour. Therefore, it is desirable to ensure that the tracker is as efficient as possible.

The mean shift algorithm [5, 6] is a nonparametric statistical method to find the nearest mode of a point sample distribution, that has been adopted as an efficient technique for image segmentation [7] and object tracking [8]. In this paper, we show how such a kernel-based object tracking algorithm can be improved by using a kernel based on the Chamfer distance [9]. We present experiments that demonstrate the superior performance of this approach in comparison with the basic algorithm by measuring accuracy, robustness and stability.

2. KERNEL DENSITY ESTIMATION

To summarise current practice, a kernel of appropriate bandwidth applied to a continuous *pdf* provides a smooth, continuous distribution that retains the original modes. Such kernels should be piecewise continuous, bounded, symmetric around zero, and monotonically decreasing from the centre [10]. We can apply such kernels to samples taken from image pixels, i.e. the kernel is defined in image, not colour space. When tracking an object through a video sequence, we represent it by a discrete distribution of samples from a region in colour space, localised by a kernel whose centre defines the current position. Then, we find the maximum in the distribution of a function, ρ, that measures the similarity between the weighted colour distributions as a function of position (shift) in the *candidate* image with respect to a previous *model* image. Defining the two sets of parameters for the respective densities as $p(x)$ and $q(x)$, the Bhattacharyya coefficient [11] is an approximate measurement of the amount of overlap,

$$\rho = \int \sqrt{p(x)q(x)}dx \tag{1}$$

As we are dealing with discretely sampled data from colour images, we use discrete densities stored as *m*-bin histograms **q**, **p(y)** in both the *model* and *candidate* image, respectively. According to the definition of Eq. (1), the sample estimate of the Bhattacharyya coefficient is given by

$$\rho(\mathbf{y}) = \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}] = \sum_{u=1}^{m} \sqrt{p_u(\mathbf{y})q_u} \tag{2}$$

In discrete space, $\{\mathbf{x}_i\}$, i=1,2,…n are the pixel locations of the model, centred at a spatial location **0,** the position of the window that we want to track in the preceding frame. A function b: $\mathbf{R}^2 \rightarrow \{1,2,\cdots,m\}$ associates to the pixel at location $\mathbf{x}_i$ the index b($\mathbf{x}_i$) of the histogram bin corresponding to the colour of that pixel. If K is the normalized kernel function, then the kernel density of the features $u = 1,\cdots,m$ in the target model estimate is given by

$$q_u = \sum_{i=1}^{n} K(\mathbf{x}_i)\delta[b(\mathbf{x}_i) - u] \tag{3}$$

where δ is the Kronecker delta function. If all possible colours in RGB space in a 24-bit image are quantised, there are 256^3 bins. As the target window is likely to be thousands of pixels at most, such a histogram would be sparsely populated. Very fine quantization of the colour space is probably unjustified for images in which the illumination may be variable, and there is additional noise on the colour video. Finally, there are several summing operations performed over the model to normalize it, which are very costly. Therefore, we use a much coarser

quantization of the colour space, 16 bins for each colour of RGB, giving a total of $m=16^3=4096$ bins.

Estimating the colour density in this way, the mean shift algorithm can be used to iteratively shift the location **y** in the target frame, to find a mode in the distribution of the Bhattacharya coefficient (Eq.2). Using Taylor expansion around the values, $p_u(\mathbf{y}_0)$, the Bhattacharyya coefficient is approximated by [8]:

$$\rho[\mathbf{p}(\mathbf{y}),\mathbf{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{p_u(\mathbf{y}_0)q_u} + \frac{1}{2}\sum_{i=1}^{n} w_i K(\mathbf{x}_i) \qquad (4)$$

where $w_i = \sum_{u=1}^{m}\delta[b(\mathbf{x}_i)-u]\sqrt{\frac{q_u}{p_u(\mathbf{y}_0)}}$

In the mean shift algorithm, the kernel is recursively moved from the current location $\mathbf{y}_0$ to a new location $\mathbf{y}_1$ according to the relation.

$$\mathbf{y}_1 = \sum_{i=1}^{n}\mathbf{x}_i w_i G(y_0 - x_i) \Big/ \sum_{i=1}^{n} w_i G(y_0 - x_i) \qquad (5)$$

where G is the gradient function computed on K. This is equivalent to a steepest ascent over the gradient of the kernel-filtered similarity function based on the colour histograms.

3. USING A KERNEL BASED ON THE CHAMFER 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 [6]. 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) [10] 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 R^2 are altered on each successive application of the algorithm. In most instances, for example in tracking the human subjects shown here, 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. Depending on the shifting of pixels between background and foreground, and on the similarity of the two colour distributions (in a worst case the background has similar properties to the target), then multiple modes are formed in the *pdf* and the mean shift is no longer exact.

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 normalised Chamfer distance transform (NCDT) rather than the true Euclidean distance, as it is an efficient approximation, as shown in Fig.1. The NCDT kernel better represents the colour distribution of the tracked target, yet retains the more reliable centre weighting of the radially symmetric kernels.

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	0	0
0	1	1	1	1	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0.0189	0.0189	0.0189	0.0189	0.0189	0.0189	0
0	0.0189	0.0377	0.0377	0.0252	0.0189	0	0
0	0.0189	0.0377	0.0252	0.0189	0	0	0
0	0.0189	0.0377	0.0189	0	0	0	0
0	0.0189	0.0377	0.0252	0.0189	0.0189	0	0
0	0.0189	0.0377	0.0440	0.0377	0.0252	0.0189	0
0	0.0189	0.0377	0.0377	0.0377	0.0252	0.0189	0
0	0.0189	0.0189	0.0189	0.0189	0.0189	0	0
0	0	0	0	0	0	0	0

(a) Binary image (b) NCDT

Fig. 1. Chamfer distances transform.

This transform is applied to the target, separated from the background by mean shift segmentation [7] or background differencing [12]. 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 NCDT transform 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 NCDT 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. Denote by s_{prev} the size in the previous frame. We measure the size s_{cur} in the current frame by running the target localization algorithm three times, with size $s=s_{prev}$, $s = s_{prev} + \Delta s$, and $s = s_{prev} - \Delta s \cdot \Delta s = 0.15 s_{prev}$. The best result, s_{cur}, yielding the largest Bhattacharyya coefficient is retained. In applying the NCDT kernel to the mean shift procedure, we have a number of options. First we can define the NCDT on the basis of the first frame, and use this for the whole sequence. Second we can update the kernel on each frame, before mean shifting in the subsequent frame but retaining the previous frame's kernel. Third, we can segment the subsequent frame and apply a different kernel weighting. This depends primarily on whether the object shape changes in the long or short term. 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 NCDT computation code is included inside the outer repeat-until loop. Otherwise, the iterative algorithm that we use to test and compare the respective kernels is the same as that defined in reference [8]:

Define target centroid, y_0, in first frame
Apply segmentation (e.g. using homogeneity criteria, background subtraction) to separate foreground (target) and background
Compute (scaled) NCDT-kernel using Chamfer distance
Form model histogram, q, in colour space
Repeat
    Fetch next frame
    Repeat

Compute candidate histogram $p(y_0)$ in colour space using NCDT-kernel

Find next location $\mathbf{y}_1$ *of candidate using Eq. (5)*

Compute error, $e = \|\mathbf{y}_1 - \mathbf{y}_0\|$

Set $\mathbf{y}_0 = \mathbf{y}_1$

Until $e \le \varepsilon$ *, an error threshold or maximum iteration reached*

$\mathbf{y}_0$ *is the new location*

Until (end of input sequence)

Adaptive kernels have also been used by Porikli and Tuzel [13]. Like our approach, their algorithm does not maintain fully the mean-shift convergence conditions [7]. However, the NCDT 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.

4. EXPERIMENTAL EVALUATION

In this section, we present the evaluation of the modified mean shift object tracking using the NCDT-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. We show examples of variation of scale and the addition of Gaussian noise. All the tests were carried out on a Pentium 4 CPU 3.40 GHz with 1GB RAM. The code was implemented in Matlab, so that it would be reasonable to assume a considerable increase in processing speed if re-implemented in another language. Even so, real-time operation is possible.

In the first experiment, we compare the tracking of a moving male pedestrian in a video sequence of a shopping centre that includes 75 frames of 320×240 pixels, comparing the normal E-kernel with the NCDT kernel. The target location was initialized by a rectangular region (shown) of size 77×31 pixels. Fig.2 shows the first frame and the foreground image of the tracked object. In this case a simple regional homogeneity criterion has been applied as the target had relatively uniform intensity. Fig.3 shows the minimum value of a distance function, $d = 2(1 - \rho(\mathbf{y}))$, computed for each frame from the Bhattacharyya coefficient (Eq. 2). By definition, the distance of the first frame is 0, meaning a perfect match. The peak in the E-kernel data is 0.643 which corresponds to the wrong candidate frame 53. After this frame the distance reduces but the algorithm was tracking another object which has nearly the same colour as the target. Fig.4(a) and (b) show some examples, frames 1, 15, 30 and 60, from the whole sequence. In frame 30 some of the original pedestrian is still contained within the window, but after the 52nd frame, the pedestrian is lost completely in Fig.4(b), as the tracker finally latches on to another crossing pedestrian. This demonstrates that the inclusion of the background of the tracked pedestrian (in this case another pedestrian) includes pixels that are similar in colour space, so that the algorithm fails to identify the correct distribution in succeeding frames and hence follows the wrong target. Fig.5 shows the manifestation of the problem in the ρ-space, using Eq. 2 – effectively the E-kernel filtered density estimate has additional, confusing modes caused by the inclusion of the crossing pedestrian in the background.

Fig. 2. Rectangular window and segmentation

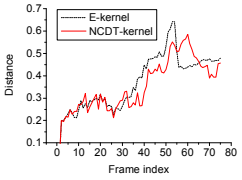

Fig. 3. The Bhattacharya distance values, for the male pedestrian

(a) NCDT-kernel

(b) E-kernel

Fig. 4. Tracking the crossing pedestrian

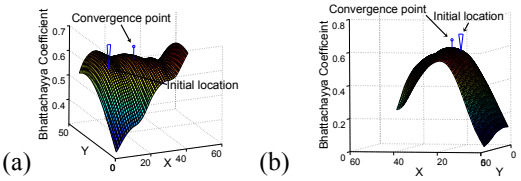

Fig. 5. The similarity surfaces (values of the Bhattacharyya coefficient) for frame 52. The initial points, (∇), and convergence points, (lines), are shown. (a) The result from the E-kernel. (b) The result from the NCDT-kernel.

In terms of complexity, computed from 200 executions of the program, the average frames per second of the NCDT-kernel and the E-kernel are 36.95 and 37.06 respectively. The maximum numbers of iterations within a single frame are 16 and 21, respectively. The average times per frame are roughly comparable because although the speed of convergence is quicker with the NCDT-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 NCDT-kernel algorithm and convergence properties, we added combined random Gaussian and uniformly distributed noise of mean zero with 0.5 and 0.05 variance respectively to the frame shown in Fig.6(a), the intensities ranging from 0 to 1. Fig.6(b) shows the one result from 1000 trials superimposed on the noisy image. For this level of noise, the algorithm is successful on all occasions, beyond this level the success rate diminishes but this data is not presented. The black rectangle is the initial position, the red one is the optimal solution. The initial position is far from the target ($\Delta x = 20, \Delta y = 50$ pixels). From Table 1, which

shows quantitative results, the NCDT kernel algorithm needs on average only 4.4 iterations to converge to the optimal result, but the E-kernel needs 21 iterations on average. Again, the greater complexity of computing the NCDT kernel is balanced by the greatly reduced number of iterations, so the processing speed per frame is comparable.

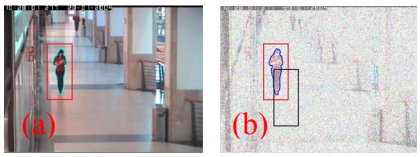

Fig. 6. (a) Original image and segmentation result. (b) Noised image and the initial rectangle.

Table 1. Comparison results of NDCT and E-kernel method

Method	Average iterations	CPU time (sec./frame)		
		max	min	mean
E-kernel	21	0.3205	0.2604	0.2801
NCDT	4.4	0.0300	0.2504	0.1815

In the third example, shown in Fig.7, the substantial difference from Fig.4 is that the intensity and colour balance of the woman is very similar to the background. Colour segmentation fails in this situation. However, we can still extract the foreground using background subtraction; we model each pixel as a mixture of Gaussians and use an online approximation to update the model [12]. The Gaussian distributions of the adaptive mixture model are then evaluated to determine which are most likely to result from a background process. Each pixel is classified based on whether the Gaussian distribution which represents it most effectively is considered part of the background model.

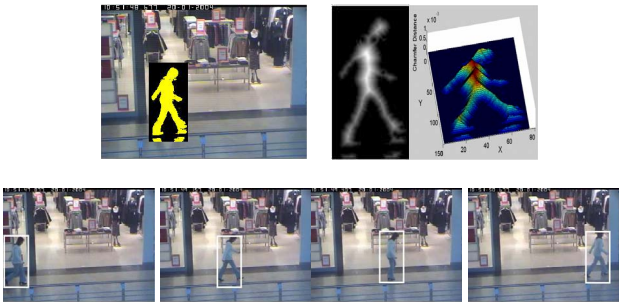

Fig. 7. On the top left is shown an example of background subtraction. On the top right are shown the 2-D and 3-D NCDT kernel, respectively. The following images are subsequent views of a female pedestrian (frames 1, 31, 51 and 70).

5. CONCLUSIONS

We have described the implementation of a scaling, normalised Chamfer distance kernel as a weighting and constraining function applied to the mean shift tracking algorithm that maximises the similarity between model and candidate distributions in colour space. In comparison with the E-kernel, used as an exemplar of a radially symmetric function, application of the NCDT-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 NCDT-kernel algorithm performs well in terms of improved stability, accuracy and robustness.

6. ACKNOWLEDGEMENTS

This work was supported by the DTC Systems Engineering for Autonomous Engineeering (SEAS) programme and the Nuffield foundation. We also used data sequences from the Caviar project at the University of Edinburgh (Fig. 4, 6 and 7) (http://homepages.inf.ed.ac.uk/rbf/CAVIAR/)

7. REFERENCES

[1] R.L. Hsu, M. Abdel-Mottaleb, and A.K. Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 24(5), pp. 696-706, 2002.

[2] R. Wildes, R. Kumar et al., "Aerial video surveillance and exploitation," *Proceedings of the IEEE*, 89(10), pp. 1518-1539, 2001.

[3] J. Deutscher, I. Reid, "Articulated body motion capture by stochastic search," *International Journal of Computer Vision*, 61(2), pp185-205, 2005.

[4] I. Haritagoglu, D. Harwood, L.S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. On PAMI*, 22(8), pp. 809-830, 2002.

[5] K. Fukunaga and L.D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, 21(1), pp. 32-40, 1975.

[6] Y.Z. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on PAMI*, 17(8), pp. 790-799, 1995.

[7] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on PAMI*, 24(5), pp. 603-619, 2002.

[8] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-based object tracking," *IEEE Transactions on PAMI*, 25(5), pp. 564-575, 2003.

[9] Gunilla Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Transaction on PAMI*, 10(6), pp. 849-865, 1988.

[10] D. W. Scott, *Multivariate density estimation*, Wiley, 1975.

[11] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematics Society,* 35, pp. 99-110, 1943.

[12] C. Stauffer, W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," *Proc. Of IEEE Conf. on Computer Vision and Pattern Recognition*, pp.246-252, 1999.

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