2010 International Conference on Computational Science and Its Applications

Real-time Non-rigid Object Tracking Using CAMShift with Weighted Back Projection

*Lei Sun
Graduate School of Information, Production and Systems
Waseda University
Kitakyushu-shi, Fukuoka-ken, Japan
*sunlei@ruri.waseda.jp

**Bingrong Wang, ***Takeshi Ikenaga
Graduate School of Information, Production and Systems
Waseda University
Kitakyushu-shi, Fukuoka-ken, Japan
**wangbingrong@ruri.waseda.jp, ***ikenaga@waseda.jp

Abstract—**The Continuously Adaptive Mean Shift algorithm (CAMShift) is an adaptation of mean shift algorithm for object tracking especially for head and face tracking. Traditional CAMShift can not deal with multi-colored object tracking and situations when similar colors exist nearby. In this paper, a new approach towards these problems using CAMShift with weighted back projection is proposed. In our approach, multi-dimensional histogram with thresholding strategy is utilized. And a new back projection weighting strategy is proposed for situations when similar colors exist near the tracked object. Through experiments, the results show that the proposed method exceeds the traditional CAMShift in situations with multi-colored object or similar-colored background while keeping the processing speed real-time.**

Keywords - object tracking; mean shift; CAMShift; weighted back projection

I. INTRODUCTION

Object tracking is a crucial task within the computer vision field. It has been widely applied in applications such as video surveillance [1], perceptual user interface [2], video coding [3] and driver assistance [4].

There are many different approaches for tracking an object [5], and mean shift algorithm is one among them aiming at real-time object tracking.

The well-known mean shift algorithm is a robust statistic method for finding the local maxima in arbitrary probability distribution. It was first proposed by Fukunaga [6], but largely forgotten until Y. Chen generalized and extended the mean shift method [7]. The mean shift method climbs the gradient of a probability distribution to find the mode. It works with a search window which is positioned over a part of the distribution and within the search window the maximum can be determined. Then the search window is moved to the position of this maximum point and new maximum is computed. This procedure is repeated until the mean shift finds a local maximum and converges. The most significant advantage is that mean shift converges very fast, which means the speed can be very fast. It is now widely used in image filtering, image segmentation, and object tracking.

CAMShift stands for Continuously Adaptive Mean Shift and it was first proposed by G. Bradski [2], aiming at efficient head and face tracking in a perceptual user interface. It is one approach to realize real-time object tracking based on mean shift algorithm. The OpenCV library [8] provides an implementation of CAMShift algorithm, which uses 1-D hue histogram and realizes adaptive scale and orientation for view-changing objects.

Another approach to realize object tracking based on mean shift is the target-candidate method, which is proposed by Comaniciu [9]. This method creates a target model and a candidate model which are usually histograms and uses a similarity function to evaluate them. And the Bhattacharya Coefficient is usually used as the similarity function.

Both the two approaches have their own merits and demerits. The CAMShift is more application (head and face tracking) oriented while target-candidate method is more of general purpose and achieves good performance for even demanding cases. CAMShift is easy to be implemented and runs very fast while target-candidate method requires slightly higher computation cost relatively. And CAMShift realizes size and orientation adaptation while the target-candidate method only provides a scale adaptation, which is not so robust.

Since CAMShift itself is not a general purpose tracker, it fails to deal with many complex situations such as multi-colored object tracking and similar color interference. In this paper, a back projection weighting strategy is proposed as a solution towards the above situations.

Traditional weighting strategies usually use Gaussian kernel or Epanechnikov kernel as the weighting function, which are easily influenced by similar colors existing near the tracked object. That's because in the mean shift implementation the weighting region is slightly larger than the tracking window which results in that the nearby colors will be considered, although the weight is insignificant relatively. In the proposed strategy, a 0-area is added which means that the pixels inside this area will be all weighted by 0. Thus the interference inside 0-area is ignored and the target will not lose easily as former methods.

The paper is organized as follows. Section II describes the detail of CAMShift algorithm. Section III describes the problems of traditional CAMShift. Proposed method to overcome these problems is explained in Section IV. Experimental results are shown in Section V and at last, conclusion is made in Section VI.

978-0-7695-3999-7/10 $26.00 © 2010 IEEE
DOI 10.1109/ICCSA.2010.39

II. CAMSHIFT ALGORITHM

The standard mean shift algorithm can only deal with static distributions (single frame) and the size of the search window is fixed. CAMShift uses a dynamic search window which is updated after every frame to realize object tracking.

The OpenCV library contains an implementation of the CAMShift algorithm using single hue histogram of HSV color space. Spatial moments are used to calculate the mass center as well as scale and orientation.

The CAMShift method can be summarized as three main procedures. First, the back projection image is calculated for the current frame. Second, mean shift procedure is applied within the calculated back projection image. And finally, the CAMShift procedure is applied to realize sequence-tracking.

A. Back Projection procedure

Back projection is a primitive operation that associates the probability of being part of the tracked object in the image of each pixel with the value of the calculated color histogram.

First, the frame is converted to HSV color space.

Second, color histogram of the target object is computed using single hue value in HSV color space of the frame. Let $\{x_i\}_{i=1...n}$ denotes the pixels in the region of tracked object, and m is the number of quantized bins. The histogram is calculated as

$$q_u = \sum_{i=1}^{n} \delta[b(x_i) - u] \,, \quad u = 1 \ldots m.$$

Here δ is the Kronecker delta function, and function b: $R^2 \rightarrow \{1 \ldots m\}$ associates the pixel at x_i with the index of its color bin in the quantized feature space.

At last, back projection image is calculated using the histogram calculated in the above step and the probability value is rescaled to be within the pixel value range. For example, for 8-bit hues, the range is between 0 and 255, as shown below

$$p_u = \min\left\{\frac{255}{\max(q)} * q_u, 255\right\}, \text{ for } u = 1 \ldots m.$$

Here the function max(q) denotes the maximum value of the histogram.

The back projection procedure flow chart is shown as follows (Fig. 1).

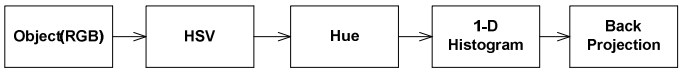

Figure 1: Back Projection flow chart

B. Mean Shift procedure

After back projection image is calculated, mean shift procedure is applied to find the mass center of the current tracked region. And the calculated mass center will be used as the initial center of the next frame. The steps are described as follows:

First, initialize the window using the hand-selected location.

Second, calculate the mass center using the statistical moments of the search window. Let I(x, y) be the intensity of the back projection image at location (x, y), then the zeroth moment and first moments for x and y are calculated as

$$M_{00} = \sum_x \sum_y I(x, y) \,,$$

$$M_{10} = \sum_x \sum_y x * I(x, y) \,,$$

$$M_{01} = \sum_x \sum_y y * I(x, y) \,.$$

Then the mass center location is calculated as

$$x_c = \frac{M_{10}}{M_{00}} \,, \quad y_c = \frac{M_{01}}{M_{00}} \,.$$

Third, move the window center to the calculated mass center.

And then repeat above steps until convergence.

C. CAMShift procedure

First, get the first frame and initialize the state.

Second, apply mean shift procedure to locate the tracked object within the frame and store the zeroth moment (it is used as the window size of the next frame) and the mass center location.

At last, set the calculated result as the initial state of the next frame and repeat. In this way, the target object in video sequences can be tracked.

The CAMShift procedure flow chart is shown as Fig. 2.

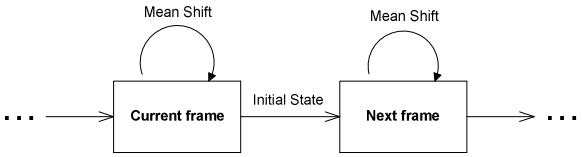

Figure 2: CAMShift flow chart

III. PROBLEMS OF CAMSHIFT

CAMShift is not a general purpose tracker since it originally targets at head and face tracking, which is mostly uniformly colored. It is suitable for uniform-colored object tracking, but it may fail to track multi-colored object.

As follows, Fig. 3 shows the back projection image when a uniform-colored football player is tracked in a football video and Fig. 4 shows the back projection image when another multi-colored football player is tracked. Whiter pixel corresponds to higher probability of being part of the tracked object. We can see that the back projection image can be a mess when the tracked object contains multiple colors.

Figure 3: Back projection image when a uniform-colored player is tracked

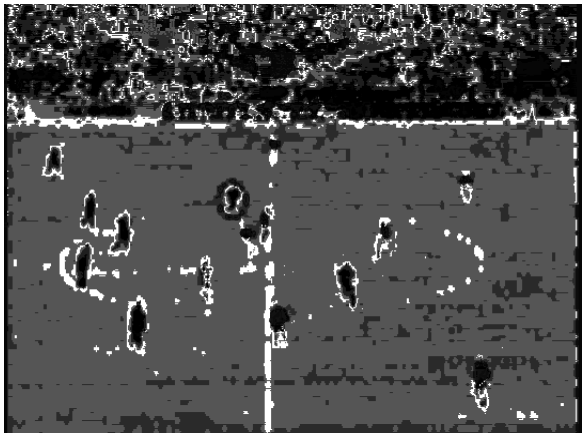

Figure 4: Back projection image when a multi-colored player is tracked

And another problem is that the hue value alone of HSV color space is not discriminative enough for multi-colored (multi-hued) scenes, although it satisfies the skin color based applications. For example, if the hue value of the tracked object is similar to the hue value of the background nearby, traditional CAMShift fails to distinguish the target object from the background, even if the object is uniformly colored.

IV. Proposed Method

A. *Thresholded multi-dimensional histogram*

Since the hue channel of HSV color space alone is insufficient to distinguish general objects, multi-dimensional histogram is used in our proposed method. And RGB color space is chosen as the feature space. In our implementation, we use 3-D RGB histogram quantized into 16X16X16 bins.

In order to ignore the insignificant colors within the track window, we set a threshold to the RGB histogram to get rid of the interference of the colors which occupy only small parts of the target region but would affect the performance significantly if not eliminated.

In our implementation, half of the maximum value of the histogram is taken as a typical threshold for general cases.

Fig. 5 shows the back projection image with the above scheme corresponding to Fig.4 above. Compared with Fig.4 which is based on the traditional CAMShift, we can see that the back projection image becomes more accurate and more concentrated with thresholded multi-dimensional histogram.

Also, whiter pixels correspond to higher possibility of being part of the tracked object.

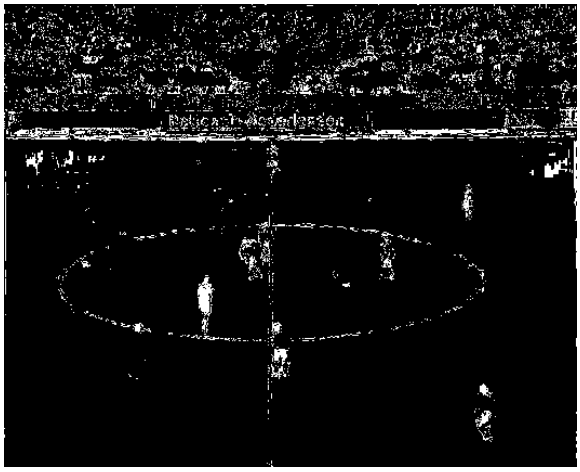

Figure 5: Back projection image with thresholded multi-dimensional histogram

B. *Weighted back projection*

In our method, the back projection image is weighted using a special strategy as shown in Fig. 6.

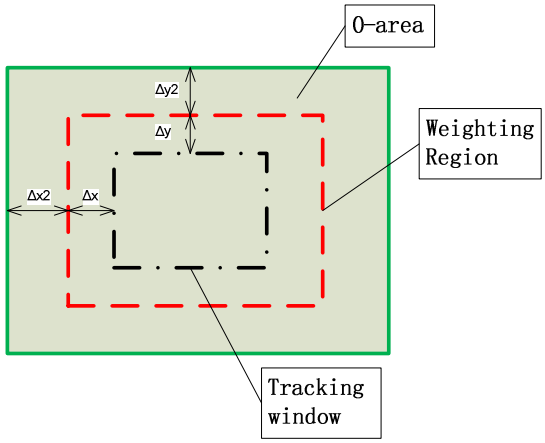

Figure 6: Weighting strategy

The innermost frame shows the tracking window where the target object is supposed to be.

The frame in the middle shows the weighting region, in which all pixels are weighted using an isotropic kernel. In our implementation, the Epanechnikov kernel as follows is used

$$k(x) = \begin{cases} 1 - x, if\ x \le 1 \\ 0,\ otherwise \end{cases},$$

where x indicates the Euclidean distance from the pixel to the center of the track window. Using this kernel, pixels far from the center are assigned with small weights, which are more likely to be background pixels.

The area between the frame in the middle and the outermost frame shows the 0-area. This area is added in order to get rid of nearby interferences from background information. Within this area, all pixels are weighted by 0.

Fig. 7 shows the result with traditional CAMShift while Fig. 8 shows the result with proposed method. The frames in Fig. 8 are hand-painted and the target in the tracking window (innermost frame) is a walking human.

We can see that with proposed method, the human can be clearly separated from the background pixels while the traditional method generates many noises around the human which lead to the tracking failure.

Figure 7: Back projection image with traditional method

Figure 8: Back projection image with proposed method

V. Experimental Results

In this section, the proposed method is applied to some sequences and the results are shown. Also, comparison with target-candidate method is shown.

In all experiments, 3-D RGB histogram with a threshold of half the maximum value is used, and it is quantized into 16X16X16 bins. Epanechnikov kernel is used as the weight function. As for the weighting strategy, the space between tracking window and weighting region frame is defined as $\Delta x = \Delta y = 7$ and the space of 0-area is defined as $\Delta x2 = \Delta y2 = 25$.

All experiments are performed on an Intel Core 2 (3.16GHZ) computer with 3.25GB RAM.

Fig. 9 shows a sequence with a woman wearing a check shirt and a pair of dark blue trousers. Besides, the skin color also exists in the tracking window. Thus the woman contains multiple colors. The results show that the proposed method tracked the woman correctly. Although the target-candidate method can also track the woman correctly, our method is smoother than the target-candidate method which is a little oversensitive.

Fig. 10 shows the walking human tracking using proposed method and Fig. 11 shows the result using target-candidate method. In this sequence, there are many noise pixels near the human like the bicycles and the statue. Fig. 10 shows that our proposed method prevented the interference of noises and achieved good results. Fig. 11 shows that the target-candidate method was influenced by the noises and finally lost the target.

However, the parameters such as Δx and Δy are crucial for applications (cases). Different applications may need different parameters.

TABLE I shows the processing time analysis of proposed method and traditional CAMShift. Since traditional method can not deal with the sequences used in Fig. 9 and Fig. 10, here I provide data of another two sequences. Column #4 shows the processing speed. Through the table, we can see that the processing speed of proposed method decreases only 0.9% and 0.2% for Football Player sequence and Human sequence respectively. And for Walking Human sequence and Woman sequence the processing speed is also kept at a high level.

VI. Conclusion

The CAMShift algorithm is simple and stable if the scene is not demanding (uniform-colored object & discriminating background). It can deal with slightly illumination and appearance change. But the performance becomes unreliable while tracking multi-colored objects or if the background interferes the tracked object. In this paper, we proposed a new method to improve the performance for this situation. Through experiments we can see that the proposed method improves the performance while keep the processing speed real-time.

ACKNOWLEDGMENT

This research is supported by the project of Core Research for Evolution Science and Technology (CREST) of the Japan Science and Technology Agency.

REFERENCES

[1] M. Greiffenhagen, D. Comaniciu, H. Niemann and V. Ramesh, "Design, Analysis and Engineering of Video Monitoring Systems: An Approach and a Case Study", Proc. IEEE, vol. 89, no. 10, pp.1498-1517, 2001.

[2] G. Bradski, "Computer Vision Face Tracking as a Component in a Perceptual User Interface", Proc. IEEE Workshop Applications of Computer Vision, pp. 214-219, 1998.

[3] A. Eleftheriadis and A. Jacquin, "Automatic Face Location Detection and Tracking for Model-Assisted Coding of Video Teleconferences sequences at low bit-rate", Signal Processing: Image Communication, vol. 7, No. 4-6, pp. 231-248, 1995.

[4] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner and W. Seelen, "Computer Vision for Driver Assistance Systems", Proc. SPIE, vol. 3364, pp. 136-147, 1998.

[5] A. Yilmaz, O. Javed and M. Shah, "Object Tracking: A Survey", ACM computing Surveys, vol. 38, No. 4, 2006.

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

[7] Y. Chen, "Mean shift, Mode seeking, and Clustering", IEEE trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 8, pp. 790-799, 1995.

[8] Intel® Open Source Computer Vision Library, 2001.

[9] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking", IEEE trans. Pattern Analysis and Machine Intelligence, vol. 25, pp. 564-575, 2003.

[10] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis", IEEE trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, 2002.

[11] Nicole M. Artner, "A comparison of Mean Shift Tracking Methods", CESCG 2008, pp. 197-204, April 2008.

[12] John G Allen, R. Xu, S. Jin, "Object tracking Using CAMShift Algorithm and Multiple Quantized Feature Spaces", Conferences in Research and Practice in Information Technology, vol. 36, 2003

TABLE I. TIME ANALYSIS

Video	#1	#2	#3	#4	#5
*Football player	30	101	1.609	62.77	15.93
**Football player	30	101	1.594	63.36	15.78
* Human	30	1832	28.56	63.83	15.67
** Human	30	1832	28.5	63.96	15.63
*Walking Human	30	451	7.031	64.14	15.59
*Woman	30	1823	28.485	63.99	15.63

*Proposed method
** Traditional method

Column definition:
#1: Frame rate of original video (fps)
#2: Total processed frames
#3: Total frame-processing time (s)
#4: Processing frame rate (fps)
#5: Average time per frame (ms)

Figure 9: Woman sequence using proposed method. The frames 167, 426, 555, 630, 703 and 879 are shown

Figure 10: Walking human sequence using proposed method. The frames 2, 100, 149 and 250 are shown.

Figure 11: Walking human sequence using target-candidate method. The frames 2, 100, 149 and 250 are shown.