

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

CamShift Based on Multi-feature Fusion and Kalman Prediction for Real-time Visual Tracking

¹Jicheng Liu, ¹Huilin Sun, ¹Hongyu Yang, ²Wencheng Li and ¹Linlin Wu

¹Laboratory of Oil and Gas Field Control and Dynamic Monitor, Northeast Petroleum University, China

²Offshore Oil Engineering Co., Ltd., China

Abstract: The tradition CamShift algorithm will prone to tracking error only with color feature. A visual tracking algorithm based on multi-feature fusion camshift combining with Kalman prediction is proposed. The algorithm with weighted color histogram and gray gradient histogram is an effective solution to the background interference. Experiments show that it can complete real-time tracking and reduce the numbers of iteration, which improves preciseness and efficiency in the tracking.

Key words: Visual tracking, camshift, multi-feature fusion, kalman filter

INTRODUCTION

Visual tracking is an important role in many applications. The most common role is for monitoring houses, parking lot, banks and other public places. The purpose of the study is to use the computer instead of human to perception, interpretation and understanding of the landscape. At present, the tracking method of moving target can be divided into four categories (Li *et al.*, 2011), namely, region-based tracking method, model-based tracking method, feature-based tracking and target tracking method bases on deformable template. CamShift algorithm is a feature matching algorithm based on color, its core is meanshift algorithm (Comaniciu *et al.*, 2003). Camshift can effectively solve the problem of target deformation and occlusion. The requirement of system resource is not high and the whole process costs less time. CamShift can achieve good tracking results in simple background. But when the background is complex or there are many similar pixels with the target's color, the algorithm will lead to failing to track (Stern and Efros, 2002; Wang and Yagi, 2006). It just considers color histogram of target, ignoring the spatial distribution of gray. So we should select Multi-feature fusion, in order to distinguish between target and background. When the target is covered, kalman filter is used to predict the position of tracking target, achieving the robustness and effectiveness of real-time tracking.

CAMSHIFT ALGORITHM

Color histogram of image is converted into color probability distribution by CamShift. We should initialize

the size and location of a search window and adaptive the position and size of the search window according to the results of the last frame. Thereby the center position of the target will be located in the current image.

The algorithm is divided into there parts which are as follows:

- **Color projection:** (back projection) (a) RGB color space is sensitive to the intensity of illumination. In order to reduce the change of illumination on tracking, the image's RGB space should be transformed into the HSV space (b) Then we should make the histogram of component H. The histogram represents the number of appeared pixels or the probability whether the values of different H component appear. In other words, we can find out the probability of component H or the number of pixels in the histogram. Then we will get the probability table of color (c) The value of each pixel in image are replaced with the probability of the appeared color. Then we will get the probability distribution of color. The process is called back projection. Color probability distribution is a gray image
- **Meanshift:** Meanshift algorithm is a non-parametric method which uses density functions with gradient estimation (Zhou *et al.*, 2010). By the iterative optimization it can find the extreme value of probability distribution to locate the target. The process of MeanShift Algorithmic: (a) Select the search window W in the color probability distribution (b) Calculating the zero-order moment:

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

One-order moment:

$$M_{10} = \sum_x \sum_y xI(x, y); M_{01} = \sum_x \sum_y yI(x, y)$$

Centroid of the search window:

$$x_0 = M_{10}/M_{00}; y_0 = M_{01}/M_{00}$$

(c) Adjust the size and wide of the search window: $S = 2\sqrt{M_{00}/256}$ (d) Moving the center of the search window to the center of mass, step (b) (c) (d) will be repeated if the moving distance is greater than the preset fixed threshold. We can stop calculating until the moving distance which between the center of the search window and the center of mass is less than the Preset fixed threshold or the frequency of loop operations reach a maximum, where (x, y) is the position of a pixel in the search window and $I(x, y)$ is the value of the pixel in the image of probability distribution.

- **CamShift:** All frames will be computed by MeanShift. Then the size and the center of search window will be gained which is used to the initial value of next frame

IMPROVEMENTS BASED ON MULTI-FEATURE FUSION

Distribution of color histogram and gray histogram: Color of an object can be used to represent the overall characteristic (Xu *et al.*, 2009). Different colors can distinguish different locations. Structure information of an object can be identified by the gradient direction that represents a trend that gray changes in an area. There will be a large difference of gradient in the edge or in the location where a large change in the gray (Wang *et al.*, 2007). Object can be reflected well by color or gradient. Color and gradient are all properties of objects. When we define an object with the two characteristics, the accuracy of the matching can be improved. Color is represented by RGB. Three sub-space (RGB) are all divided into 16 parts. There are 16 colors in every sub-space. Then an object contains 48 sub characteristics. The gradient space is divided into 16 levels by the similar processing. Each level is 22.5 degrees. Combining the gray gradient histogram and color histogram, we get the joint histogram. Thus there are 64 levels in the joint histogram. According to MeanShift algorithm, the multi-feature histogram distribution of candidate target is as follow (Lu *et al.*, 2010).

$$\hat{p}_u(y_i) = \begin{cases} c \sum_{i=1}^n \left[k \left\| \frac{y_j - x_i}{h} \right\|^2 \right] \delta[\lambda_1(x_i) - u], & 1 \leq u \leq 48 \\ c \sum_{i=1}^n \left[k \left\| \frac{y_j - x_i}{h} \right\|^2 \right] \delta[\lambda_2(x_i) - u], & 49 \leq u \leq 64 \end{cases} \quad (1)$$

where, λ_1 is the eigenvalue for the color space of pixel x_i and λ_2 is the eigenvalue extracted from the pixel's gray. At the same time u is the level and c is the normalization constant. y_j is the center position of the target, h being the bandwidth of the kernel function k .

Visual tracking is to search a region which must match the color histogram of target model and the gray gradient histogram of target. So, tracking error will appear when the feature can not be distinguished in the same or similar color. Because each different area has different gradient directions in the image, the tracking error of Meanshift algorithm caused by the interference from same color need not to be considered.

Weighted multi-feature and feature extraction: When the tracking target is obstructed partially, we should statistic the proportion of A to B (A is the number of pixels fallen in the each color space or the gradient direction characteristic intervals, belonging to the area around the target; B is the total number of pixels in the target area). Thus we will get the characteristic probability distribution of the area around target. The surrounding area contains the more pixels of a feature, the greater the probability of this feature is. It is easy to generate an error when we use this feature to indicate the target. Actually, the most representative characteristics or characteristic regions which can distinct between target and background easily should be given more weighted. In this way, robustness can be ensured when the target is obscured. The target and background are always mixed together. We have no way to distinguish clearly. When we use MeanShift algorithm to modeling by a rectangular window for the target and the candidate's target, the rectangular window contains not only the target but also background. This situation will lead to model distortion. When the target is not obvious, target and background occupy almost the same proportion in this rectangular window. The histogram contains a lot of background information, resulting in erroneous tracking. So, we should divide the histogram into several small intervals. The interval is called bin. Through repeated experiments, we summarize that the logarithmic ratio between target and background represents bin's ability to distinguish target from background. We just need to find a big difference between the target and its immediately adjacent neighborhood. Assuming $A = \{a_v\}_{v=1, \dots, m}$ (The normalized histogram of target area) and $B = \{b_v\}_{v=1, \dots, m}$ (The

normalized histogram of background area) They each contain m bin. There is a small real number ξ . Then the logarithmic ratio of the N -th bin is:

$$z = \log(\max(a_n, \xi) / \max(b_n, \xi)), n = 1, \dots, m \quad (2)$$

Histogram A is weighted with the weight Z . In this way, histogram reflects the different bin's ability to distinguish the target from background. We assume that the original histogram is L . The weighted histogram of logarithmic ratio is:

$$L' = \{L'_v\}_{v=1, \dots, m} L'_v = L_v \cdot z', v = 1, \dots, m \quad (3)$$

where, L'_v represents the weighted value of the V th bin, z' is normalized z .

Obviously, if z is small, the corresponding bin will appear more often in the background but less in the target. By the above steps we find that the improved histogram of logarithmic ratio can accurately reflect the ability to distinguish between background and target. We can clearly see the difference between them. Logarithm of the ratio is good for avoiding other bin invaliding when the weight of a bin is too large.

IMPROVED ALGORITHM

The traditional CamShift can effectively solve partial occlusion. But when the target is severely obscured, Kalman filter to be added to predict the target motion parameters, in order to prevent the loss of tracking targets. We will calculate according to the following steps without severe occlusion.

- **Detecting moving targets:** Automatically obtaining target position in the initial frame and calculating the weighted color histogram and the weighted gray histogram. We initialize the target model, candidate area and the center position y_0 of candidate region (1) calculating H component histogram of target region; (2) initializing the candidate area, y_0 is the center coordinates of the candidate region
- Reading the next frame, using the target center of the last frame as the present center position. The half length and width of target minimum bounding rectangle as the bandwidth of the search window, computing the target candidate weighted color histogram and target weighted gray gradient histogram, namely back projection

- **Calculating a new candidate position y_0 of target:** (1) Calculating the new target position y_1 ; (2) Let $d = \|y_1 - y_0\|$, $y_0 = y_1$. Assuming that ϵ is Error threshold, N is Maximum number of iterations. When $d < \epsilon$ or $k \geq N$, iteration ends. Then program returns to step 2. y_1 as the observed value to update kalman filter. Otherwise, $k = k + 1$, jump to 1)

When the target is severely covered, the predictive value of kalman filter is used as observed value, to update the kalman filter. Then program continues according to the above step. We can determine whether the target with severely occlusion by $r(k)$:

$$r(k) = \sqrt{(x(k) - \hat{x}'(k))^2 + (y(k) - \hat{y}'(k))^2} \quad (4)$$

$$\text{Rect_current} < \beta \times \text{Rect_origin}, \beta \in (0, 1) \quad (5)$$

where, $x(k)$, $y(k)$ as the position of observed value. $\hat{x}'(k)$, $\hat{y}'(k)$ as the position of predictive value. If $r(k) > \alpha$ (α is an adaptive threshold. However, according to experiments, we summarize that α is 10) and $\text{Rect_current} < \beta \times \text{Rect_origin}$, $\beta \in (0, 1)$, we should think the target with severe occlusion. Rect_current is the size of the calculated target by CamShift algorithm. Rect_origin is the size of initial target.

EXPERIMENTAL RESULTS AND ANALYSIS

The first experiment is a vehicle tracking in traffic scene. The tracked vehicle is traveling on the road with a lot of traffic and the car is often obscured by other vehicles. There are 300 frames in this test sequence. The size of each frame is 352×240 . And the size of the tracked vehicle is 28×20 . Figure 1 shows the vehicle tracking result in traffic scene.

It can be seen that the algorithm can achieve robustness tracking in vehicle traffic scenarios from the figure above. Particularly, it still track steady when part of the tracked vehicle has been obscured. We can accurately identify the target by gradient histogram and easily solve the problem of occlusion with kalman filter.

Figure 2 is a real-time tracking of the target. There is a holster on the fingernail. We can select the holster as the target and locate target on the fingernail. By the improved algorithm that we have introduced we can distinguish and track the target with the fingernail keeping moving. The improved algorithm can still achieve robustness tracking with the interference of light by the color histogram and gray gradient histogram.

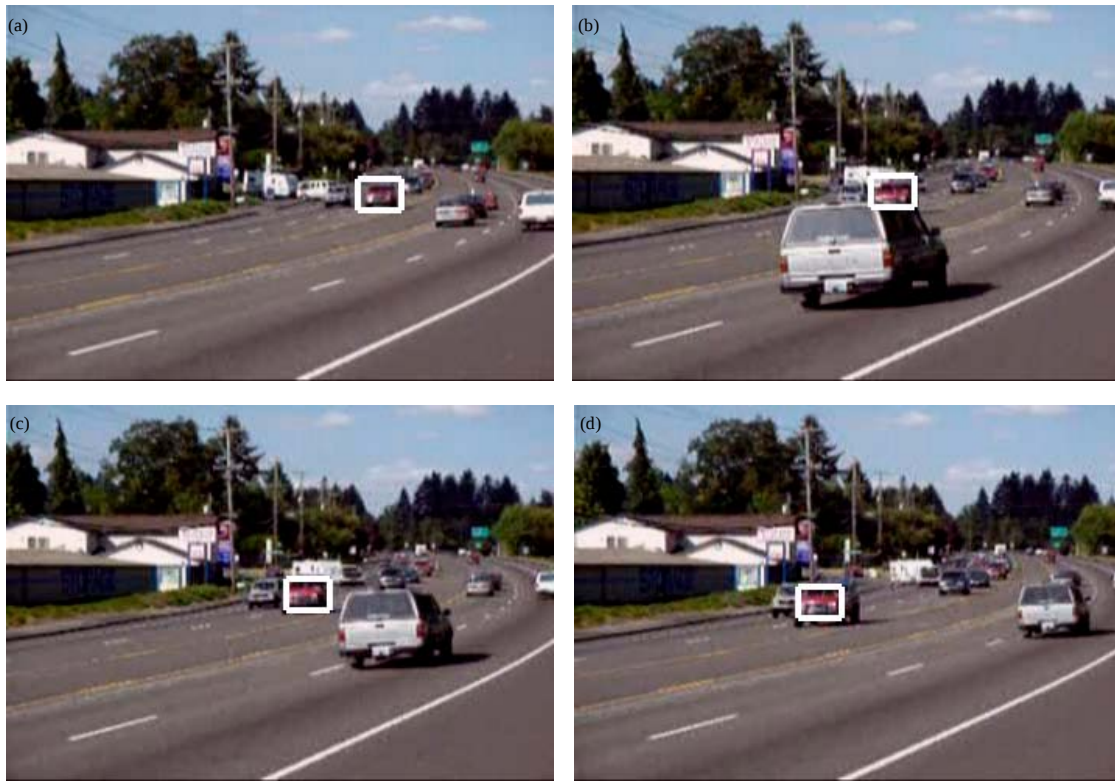


Fig. 1(a-d): Visual chart of vehicle tracking

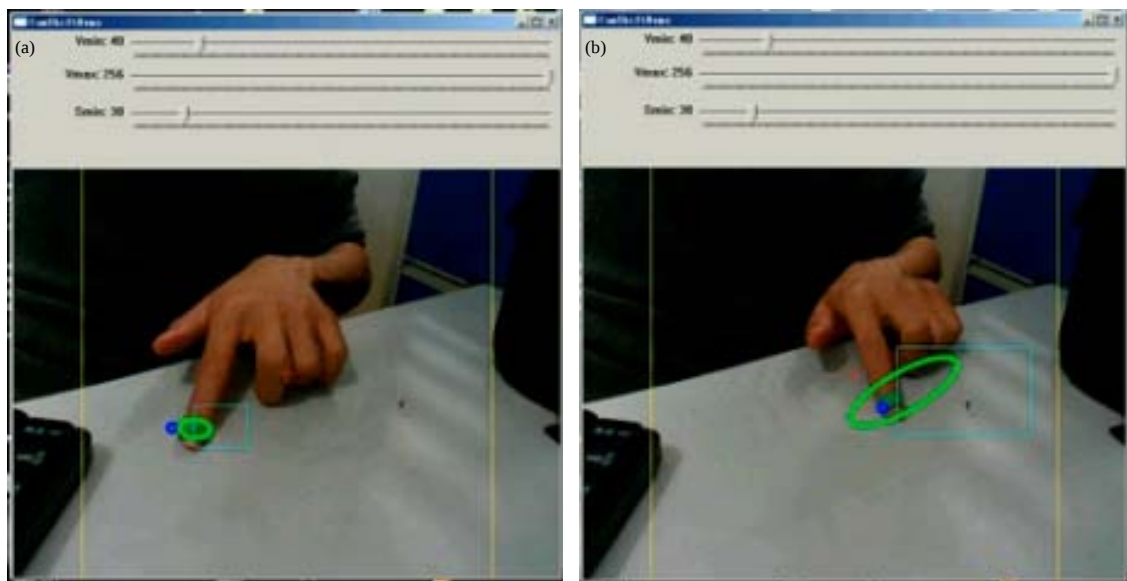


Fig. 2(a-f): Continue



Fig. 2(a-f): Real-time target tracking process

Table 1: Comparison between traditional algorithm and improved algorithms in the same experimental scene

Algorithm	Processing time	Average number of iterations
Traditional algorithm	0.083	2.6
Improved algorithms	0.081	2.0

The traditional algorithm and the improved algorithm all meet the general requirements of tracking. But the improved algorithm has an advantage in real-time tracking. Table 1 shows that average processing time per frame.

The Table shows the processing time is almost the same. But the iterations of the improved algorithm is

fewer. Kalman filter can confirm the direction of next search area and reduce the size of the area when we use Camshift algorithm. That is to say, the number of iterations must be reduced. When there are massive frames in the video, the improved algorithm will increase efficiency.

CONCLUSION

In this study, we propose a visual tracking algorithm that CamShift based on multi-feature fusion combines

Kalman prediction. It can splendidly complete robustness tracking and real-time tracking. The target is identified by the color histogram and gray gradient histogram, achieving the robust tracking. Kalman filter is added to predict the target position. Effective tracking can still be achieved when the target has been severe obscured in this way. Thus robustness of the system will be improved. And the number of iterations is shortened by about 23%, greatly improving the efficiency of visual tracking.

ACKNOWLEDGMENT

The study is sponsored by Chang Jiang Scholar Candidates Programme for Provincial Universities in Heilongjiang (CSCP).

REFERENCES

- Comaniciu, D., V. Ramesh and O. Meer, 2003. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 25: 564-577.
- Li, M., L. Jing and J. Zou, 2011. Combined with extended kalman filtering CamShift moving target tracking algorithm. *Lightning Control*, 4: 2-5.
- Lu, X., H. Lei and Z.B. Hao, 2010. Automatic CamShift tracking algorithm based on multi-feature. *Comput. Appli.*, 30: 650-652.
- Stern, H. and B. Efron, 2002. Adaptive color space switching for face tracking in multi-colored lighting environment. *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition*, May 20-21, 2002, Washington, DC., USA., pp: 249-254.
- Wang, J. and Y. Yagi, 2006. Integrating shape and color features for adaptive real-time object tracking. *Proceedings of the International Conference on Robotics and Biomimetics*, December 17-20, 2006, Kunming, pp: 1-6.
- Wang, Q., S.H. Jiang, J.G. Zhang and B. Hu, 2007. An approach to improve the performance of mean-shift tracking algorithm. *J. Fudan Univ.*, 46: 85-90.
- Xu, K., Y.Y. He and W.Y. Wang, 2009. The adaptive color space target tracking algorithm based on CamShift. *Comput. Applic.*, 29: 757-760.
- Zhou, S., P. Hu and Y. Liu, 2010. Based on the improved mean shift and adaptive kalman filtering video target tracking. *Comput. Appli.*, 30: 1573-1576.