Real-Time Both Hands Tracking Using CAMshift with Motion Mask and Probability Reduction by Motion Prediction

Ryosuke Araki* Seiichi Gohshi† and Takeshi Ikenaga*
† Kogakuin University, Tokyo, Japan
* Waseda University, Tokyo, Japan
E-mail: ravelle-music@suou.waseda.jp Tel/Fax: +81-3-5286-2350

Abstract—**Hand gesture interfaces are more intuitive and convenient than traditional interfaces. They are the most important parts in the relationship between users and devices. Hand tracking for hand gesture interfaces is an active area of research in image processing. However, previous works have limits such as requiring the use of multiple camera or sensor, working only with single color background, etc. This paper proposes a real-time both hands tracking algorithm based on "CAMshift (Continuous Adaptive Mean Shift Algorithm)" using only a single camera in multi-color backgrounds. In order to track hands robustly, the proposed algorithm uses "motion mask" to combine color and movement probability distributions and "probability reduction" for multi-hand tracking in non-limiting environments. Experimental results demonstrate that this algorithm can precisely track both hands of an operator in multi-color backgrounds and process the VGA size input sequences from a web camera in real time (about 25 fps).**

I. Introduction

Computers have become common in our daily lives, and many more people use them on a regular basis. People have been using keyboards and mouse devices for many years. Recently, touch panels have become common and they do not need keyboards or mouse devices. However, we need simpler interfaces to use computers and some devices. Man-machine interfaces are one of the key technologies to exploit computers. Many methods have been proposed in man-machine interface fields. Hand gestures are one of the interesting candidates among them since they are intuitive and convenient actions. They are capable to control computers remotely. Gesture recognition systems like Kinect have become practical and have spread in our lives. The research field of hands tracking and recognition is one of the active research areas in the image processing.

Hand tracking and recognizing are the common issues. Many approaches to track or recognize hand position are developed today. Wang et al. [1] proposed the method for tracking hand and getting hand shapes by combination of colorful gloves and a camera. This work can track hands and detect hand shapes. Additionally, this work can detect 3D motion precisely in real time. They use this method for the interfaces of CAD systems and so on. Elmezain et al. [2] proposed the way to track hands by stereo camera and recognize hand gestures by Hidden Markov Models (HMM) that is widely used for voice and gesture recognition. This way can get the user's face and the location of both hands in a three-dimension, and recognize hand gestures by orientations of moving directions. Chen et al. [3] used a machine learning technique, Haar-Like feature, to get hand positions and postures, and Maung [4] used neural network for detecting hand shapes. These researches can recognize hand shapes that are detected the prepared gestures precisely, and control some devices by these gestures. Nadgeri et al. [5] used CAMshift algorithm for hand tracking. Their algorithm is based on orientation histogram for feature vectors to get the same hand at different frames and positions. This method can track the same hand shapes in the input sequences.

However, these previous works have several limitations such as requiring the use of multiple camera, sensor or hand glove, working only with single color backgrounds or practical hand shapes. These are the limitations of man-machine interfaces, too. For a fast and precise hand tracking method, this paper focuses on "CAMshift" algorithm [6]. CAMshift can track target windows with simple processing, and it is possible to get tracking target positions in real time. This method only finds peaks of probability distributions to track targets. The advantages of CAMshift algorithm are robustness to irregular object motion, image noise and partial occlusion and fast processing. However, several issues exist in the previous methods using only color data for probability distributions. To avoid these issues, this paper proposes "motion mask" to combine color and movement data, and "probability reduction" to track multiple targets.

II. CAMshift Algorithm

The conventional CAMshift algorithm applies color data to probability distribution, especially hue data in HSV. The color probability distribution map called backprojected image is made and used for tracking operations. CAMshift is the same tracking technique as "Mean shift" algorithm [7]. Mean shift algorithm is a non-parametric technique that climbs the gradient of a probability distribution to find the nearest dominant mode (peak). The operating flow is following steps.

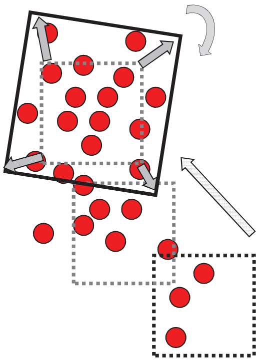


Fig. 1. Concept image of CAMshiftm.

The size and the location of initial searching windows are detected, and computed the mean location (centroid) in the window. To compute centroid, zeroth moment M_{00} and first moments M_{10}, M_{01} are calculated as follows.

$$M_{00} = \sum_x \sum_y I(x, y) \tag{1}$$

$$M_{10} = \sum_x \sum_y xI(x, y), M_{01} = \sum_x \sum_y yI(x, y) \tag{2}$$

In these equations, $I(x, y)$ means the pixel or probability value in point (x, y). A centroid of the search window (x_c, y_c) is obtained as follows.

$$(x_c, y_c) = (\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}) \tag{3}$$

The center of the search window moves to centroid and these operations are repeated until less than a minimum movement or more than maximum cycle times.

Additionally, CAMshift calculates length, width, and angle of the search window by second moments. Fig. 1 is a concept image of CAMshift algorithm from initialization to the end of tracking operations.

III. Proposed Method

The conventional CAMshift had some issues by these operations. First of all, when some objects in the background have the similar probability distributions with the target, there are frequently some tracking errors. Secondly, as several targets are tracked, CAMshift can't recognize approaching targets separately. This paper proposes following three steps to improve them.

1) Getting color probability distributions from a face
2) Getting motion vectors and creating a motion mask
3) Setting probability reduction areas from the other windows and motion prediction

A. Getting color probability distributions from a face

The color probability distribution is sensitive to light conditions and positions of the targets. Fixed threshold values are not sufficient for getting a skin color mask to pick up hands from input sequences. To improve this issue, the basis hue value of skin color is decided by a user's face area extracted automatically by haar like feature [8]. This method can get better threshold values of skin color in various situations and create the color probability distribution mask precisely.

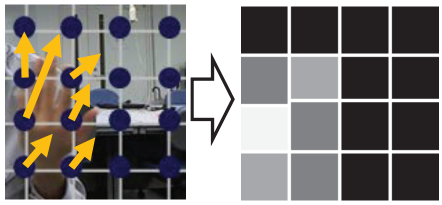
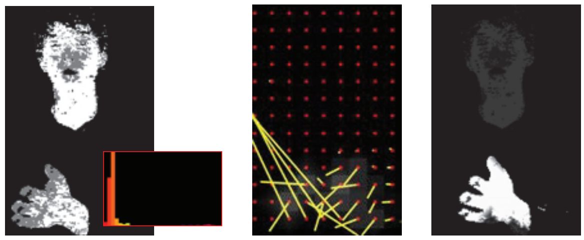


Fig. 2. Motion mask using motion vector.



(a) color (b) mask (c) feature

Fig. 3. Probability distribution image combined color and motion.

B. Getting motion vectors and creating motion mask

The previous approach can extract the strict color probability distribution masks from input images. However, this mask image has some noise because of skin color objects. The conventional CAMshift cannot track hands correctly in this situation. To resolve this issue, motion mask to use motion vector for additional probability distribution is proposed.

Starting points of vectors are extracted in grid patterns. Motion vectors are computed by pyramid LK algorithm, one of the optical flow algorithms like Fig. 2. The value of a probability distribution f_i around the starting point i is calculated by the following equation.

$$f_i = \begin{cases} v_x + v_y, & \text{if } v \leq threshold \\ 0, & \text{if } v > threshold \end{cases} \tag{4}$$

where v_x and v_y mean the value of the x and y directional vector.

After getting the motion mask, the last probability distributions, "$feature$" is calculated by color and motion probability distributions for tracking operations.

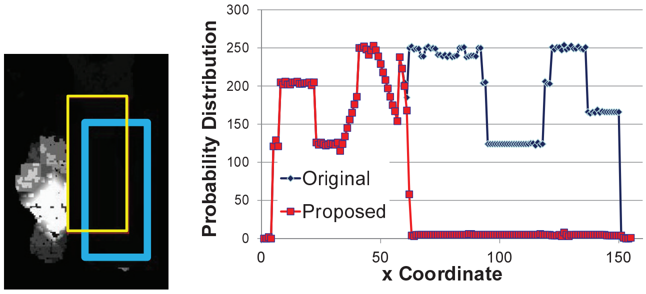
$$feature = \alpha * color + \beta * mask \tag{5}$$

In (5), "$color$" means the value of the color probability distirbution, "$mask$" means the value of the motion mask. α and β mean constants, and their values are $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$.

Additionally, initial search windows are selected from the area that vector values exceed threshold. This way can select an initial window automatically in moving hands.

C. Setting probability reduction areas from the other windows and motion prediction

CAMshift algorithm finds the nearest peak to track targets in the probability distribution. However, when tracking targets are approaching each other, the peaks are combined and become a wide peak. In this case, CAMshift cannot recognize each hand separately, and this becomes cause of the tracking error. This paper proposed probability reduction areas.



(a) Reduction area (b) Graph of probability distribution

Fig. 4. Probability reduction areas by previous and predicted positions.

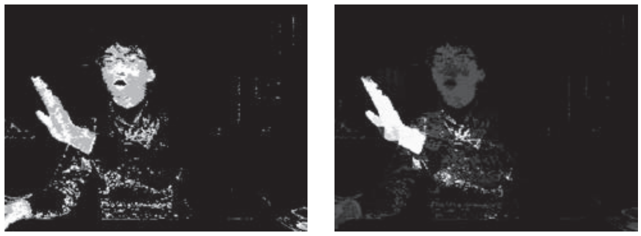
In this method, when the target is tracked by CAMshift in a current frame, the probability distribution in the previous positions of other windows is cut to less the constant value. This operation can avoid the cause to combine the target peaks. This method reduces the probability distributions in previous target positions of the others, however fast and big moving targets are not included in these areas. So, probability reduction areas are expanded to some areas predicted by Kalman filter. Kalman filter is an estimate algorithm of the minimum variance for status switch in the dynamic system. It can predict the position and the velocity of the linear moving targets. Fig. 4(a) is an image of probability reduction areas when a right hand is tracked. Around a left hand, a thick blue rectangle is a previous place of CAMshift tracking, and a thin yellow rectangle is a prediction place by Kalman filter. The left hand is comprised in previous and predicted positions in this image, and the right hand is emphasized through the influence of motion mask. Fig. 4(b) is a graph image of the probability distribution. The position of image is the center line of Fig. 4(a). The original line has a wide peak area around both hands, so CAMshift cannot recognize each hand. On the other hand, the proposed line only has a peak around the right hand, and the probability distributions in the left hand and the prediction areas are decreased.

IV. Experimental Results

To compare the conventional CAMshift and our proposed method, some test sequences taken in non limited background are prepared by a web camera. The size of the test sequences is VGA (640×480) size, and the condition of these experiments is notebook that CPU is Intel Core 2 Duo 2.53 [GHz], and memory is 4 [GB].

A. Precision comparison of hand tracking

In first experiment, the tracking accuracy is compared by the test sequence that a left hand moves in front of the face with different speed. A part of result images is shown in Fig. 5, and Fig. 6 is the probability distribution image that CAMshift use for tracking. As shown in Fig. 5(a), the traditional CAMshift algorithm cannot track the hand when the tracking window passes the face. This often occurs after moving quickly or becoming small shapes because of the face that has the similar color probability distributions. In Fig. 6(a), the areas of a face and a hand are the similar probability distributions in the conventional CAMshift using only color. On the other hand, the proposed method can track the hand continually after passing over the face. As shown in Fig. 6(b), the moving hand position is emphasized clearly by motion mask.



(a) Traditional CAMshift (b) Proposed method

Fig. 6. Probability distributions of the result image in frame 26.

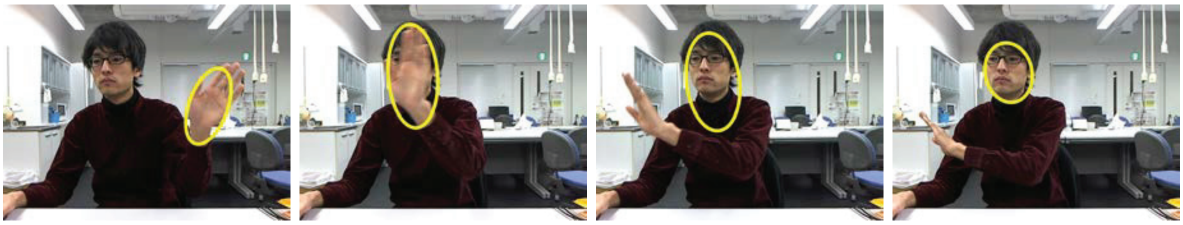
Secondly, the movie that both hands are approached each other several times is used for precision comparison. As shown in Fig. 7(a), if the both hands are distant, the traditional method can recognize each hand. However, when both hands are approached, tracking windows are unified by the combination of probability distribution peaks. This unified window is not separated after this frame. This is the characteristic of the conventional CAMshift. The proposed method in Fig. 7(b) can detect each hand continuously after the both hands are approached. When one hand is tracked, the probability distributions in the other hand and the predicted positions are reduced by the probability reduction method. Our method can track and recognize each hand accurately since the both hands are moved with different speed, shape and size.

B. Processing time

In second experiment, average processing times are monitored by some test sequences. Proposed method can parallelize operations to get the color and motion probability distributions, so parallel processing results are also indicated. These results are displayed in Table I. In this table, the traditional CAMshift algorithm is faster than the proposed method because of the simplest operations. The traditional method can process over 30 fps. The processing time of the proposed method is about 5 times bigger than the original. However, this fps is over about 25 fps, and the input images from a web camera are processed in real time by notebook. The half time of these processing times is occupied by the process to get motion vectors. If the grid size is more distant than these experimental size (30×30), the processing times are faster than this result. When parallel processing is used, this result is better than non parallel processing. From these processing time data, our method has the probability of applying man-machine interfaces for some devices.

V. Conclusions

In order to track the both hands of an operator accurately from the several hand gestures in real time, this paper proposes motion mask to combine the probability distributions of the color and motion, and probability reduction areas of the other target positions and the motion prediction to track multiple targets. From several experiments, the proposed method can



(a) Traditional CAMshift

(b) Proposed method

Fig. 5. Result images in background having similar color objects. (from left, frame 17, 22, 26, 30)

(a) Traditional CAMshift

(b) Proposed Method

Fig. 7. Result images that each hand approaches. (from left, frame 504, 511, 512, 516)

TABLE I

AVERAGE PROCESSING TIME

Sequences Name	Total Frames	Traditional	Proposed	Proposed (Parallel)
Sequence crossing the face 1	150	8.5 ms	40.7 ms	32.8 ms
Sequence crossing the face 2	309	8.97 ms	41.2 ms	32.6 ms
Sequence approaching each hand	300	8.99 ms	41.8 ms	33.9 ms
Sequence approaching and crossing 1	341	8.99 ms	40.1 ms	31.6 ms
Sequence approaching and crossing 2	647	8.6 ms	41.0 ms	33.0 ms

track and recognize hand positions in multi-color environments that the user faces a single camera, and this method is available for man-machine interface.

ACKNOWLEDGMENT

This work was supported by KAKENHI(23300018).

REFERENCES

[1] R. Y. Wang and J. Popovi, "Real-Time Hand-Tracking with a Color Glove," *ACM SIGGRAPH 2009 papers,* pp. 63:1-8, 2009.

[2] M. Elmezain, A. AI-Hamadi, lAppenrodt, and B. Michaelis, "A Hidden Markov Model-Based Continuous Gesture Recognition System for Hand Motion Trajectory," *Proceedings of the 19th International Conference on Pattern Recognition(ICPR' 08),* December 2008.

[3] W. H. Chen, Y. H. Lin, and S. J. Yang, "A generic framework for the design of visual-based gesture control interface," *IEEE Conference on Industrial Electronics and Applications (ICIEA),* pp. 1522-1525, 2010

[4] T. H. H. Maung, "Real-time hand tracking and gesture recognition system using neural networks," *World Academy of Science, Engineering and Technology 50 2009,* pp. 466-470, 2009

[5] S. M. Nadgeri, S. D. Sawarkar, and A. D. Gawande, "Hand gesture recognition using CAMSHIFT algorithm," *International Conference on Emerging Trends in Engineering and Technology (ICETET),* pp. 37-41, 2010

[6] G. R. Bradski, "Real Time Face and Object Tracking as a Component of a Perceptual User Interface," *Proc. IEEE Workshop Applications of Computer Vision,* pp. 214-219, 1998.

[7] Y. Cheng, "Mean Shift, Mode Seeking, and Clustering," *IEEE Trans. Pattern Anal. Machine Intell,* pp. 17:790-799, 1995.

[8] P. Viola and M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2001.

[9] R. Araki and T. Ikenaga, "Real-time Hands Tracking Using Feature Point Gathering Based on KLT Tracker for Man-Machine Interface," *APSIPA Annual Summit and Conference,* Oct. 2011.