REAL-TIME HAND LOCATING BY MONOCULAR VISION

Li Ding, Jiaxin Wang
Department of Computer Science and Technology, Tsinghua University, Beijing, China

Christophe Chaillou
ALCOVE, INRIA Lille Nord Europe, Lille, France

Chunhong Pan
LIAMA, Institute of Automation, Chinese Academy of Sciences, Beijing, China

Keywords: Hand Tracking, Object Occlusion, Possibility Support Map.

Abstract: The research on real-time hand locating by monocular vision has a considerable challenge that to track hands correctly under occlusion situation. This paper proposes a robust hand locating method which generates a possibility support map by integrating information from color model, position model and motion model. For better accuracy, hands are modeled as ellipses. The PSM depends on both previous model information and the relationship between models. Hand pattern search is then processed on the generated map by two steps which firstly locates the center position of hand and secondly determines the size and orientation. Our experimental results show that the proposed method is efficient under situation that one hand is occluded by the other one. Our current prototype system processes image at 10~14 frames per second.

1 INTRODUCTION

In recent decade, more and more Human Computer Interaction(HCI) applications apply vision-based interaction input methods where users can directly give instructions by natural gestures instead of typing commands on a keyboard or manipulating a mouse. Quantities of applications and systems retrieve information by monocular vision: some of them need only 2D information and others require a 3D approach (for example virtual avatar control). Hand locations are mostly computed, because we routinely use upper limbs especially hands gestures when we describe or direct during communications/ interactions.

The systems described in (Tamura and Kawasaki, 1998) and (Starner and Pentland, 1995) focus on tracking one hand for sign language recognition. These systems find and track a hand from a video image sequence and analyse its shape and motion. As they are only able to track single hand, they become ineffective when hand occlusion happens. They either throw a warning to user or ignore the overlapping period. However, in practical situations, user often move their hand in front of the face or the other hand. This observation motivates us to develop a robust method for locating hands under occlusion in real-time for HCI application by monocular vision.

In this paper, we present a real-time hand locating system which tracks the hands of a person who is driving his virtual avatar by his gestures. The system locates the hands even if they are occluded by each other. The next section introduces some related hand locating approaches and systems. In section 3, we will show the generation procedure of Possibility Support Map (PSM) which is the basis for a further two-step tracking procedure (described in section 4). Some experimental results are shown in section 5.

Ding L., Wang J., Chaillou C. and Pan C. (2010).
REAL-TIME HAND LOCATING BY MONOCULAR VISION.
In *Proceedings of the International Conference on Computer Vision Theory and Applications*, pages 406-412
DOI: 10.5220/0002816304060412

2 RELATED WORK

There are mainly two categories of methods to locate hands in a video sequence. One of them is based on skin region detection. The approach in (Hasanuzzaman, 2004) detects skin region in YUV color space, and uses x coordinate to extinguish left and right hands. They classify the extracted hands locations into 8 command gesture patterns in order to operate a robot. An approach based on region SVM learning is proposed in (Han, 2006), which automatically segment skin regions out of a video frame and assume the largest three regions as head and hands. Both of these approaches are effective locating two hands under the condition that skin regions are isolated to each other. In order to distinguish two hands when they get close, several attempts are made. In (Lee and Cohen, 2006), blob merging technique is introduced to locate the forearm (therefore locate the hand) in static images. The skin regions of forearm and hand are modeled as ellipses under the assumption that user wears shorts. Askar et al. (Askar, 2004) tried to handle the situation with contact between hands. The approach depends on row and column histograms analysis of the mask binary image, therefore they may fail due to the noises derived from background suppression.

The other category is based on object tracking techniques. Meanshift(Comaniciu and Ramesh, 2000) and Camshift(Bradski, 1998) are widely used in tracking single object. However these two methods could not handle the situation of multiple objects (two hands and probably head). The methods will lose object or mislead to track inappropiate object when objects are getting close to each other. In (Vacavant and Chateau, 2005) particle filter is used to find proper positions of head and hands. It performs at 6 frame per second, which is not suitable for real-time application. Hand occlusion is the main challenge in hands tracking, because not only the overlapping often occurs, but also the hand size and the hand orientation are both changing during occlusion. Many systems like (Schreer, 2005), (Kirubarajan and BarShalom, 2001), (Coogan, 2006) and (Imagawa, 1998) will warn the user to separate hands when occlusion is detected and the systems will restart tracking after then.

Real-time hands locating, especially when occlusion happens, is still a tempting research area. In our system, we firstly generate PSM with three aspects of hand information, and then apply tracking procedure on the PSM.

3 POSSIBILITY SUPPORT MAP GENERATION

For each frame, a multi-channel PSM is generated to support the further tracking procedure. The channel number of PSM is set to the quantity of objects which are being tracked. In our case, we track three objects: two hands as well as head. Each channel of the map is generated from original image and every pixel in it represents the possibility of coresponding object. The possibility that a pixel supports the object is calculated by combining color information, position information and motion information.

3.1 Color Information

Although colors captured by the camera are in RGB mode, they are converted into HSV mode, since color footprint is more distinguishable and less sensitive to illumination changes in the hue saturation space. Therefore color C could be represented by (C_h, C_s). For the pixel in color C, the possibility of being in the skin region is denoted as p(skin|C) and calculated by equation 1which is derived from Bayesian Equation.

$$p(skin \mid C) = \frac{p(C \mid skin) \times p(skin)}{p(C)} \quad (1)$$

In our situation, the assumption that every pixel has the same possibilities p(C), p(skin) and p(C|skin) is made, so that following equations can be used.

$$p_t(skin) = \frac{N_t(skin)}{N_t} = \frac{\sum_{i=1}^{t} n_i(skin)}{\sum_{i=1}^{t} n_i} \quad (2)$$

$$p_t(C) = \frac{N_t(C)}{N_t} = \frac{\sum_{i=1}^{t} n_i(C)}{\sum_{i=1}^{t} n_i} \quad (3)$$

$$p_t(C \mid skin) = \frac{N_t(C \mid skin)}{N_t(skin)} = \frac{\sum_{i=1}^{t} n_i(C|skin)}{\sum_{i=1}^{t} n_i(skin)} \quad (4)$$

Where $n_i(C|skin)$ is the number of skin region pixels in color C; meanwhile $n_i(C)$, $n_i(skin)$ and n_i are respectively the number of pixels in color C, the number of skin region pixels and total pixel number for the i^{th} frame.

If the skin region portion in the consecutive frames hardly changes, Equation 2~4 can be derived as follow.

$$p_t(C \mid skin) \approx (1-r)\, p_{t-1}(C \mid skin) + r \frac{n_t(C \mid skin)}{n_t(skin)} \quad (5)$$

$$p_t(C) = (1-r)\, p_{t-1}(C) + r \frac{n_t(C)}{n_t} \quad (6)$$

$$p_t(skin) = (1-r)\, p_{t-1}(skin) + r \frac{n_t(skin)}{n_t} \quad (7)$$

These equations are used to update possibilities after each frame, and the factor r is called learning rate which is set to 0.05 in our system. The larger r is, the faster the system adapts to currently captured image.

The color likelihood function is defined as

$$p_{color}(P) = p(skin \mid C) \quad (8)$$

3.2 Position Information

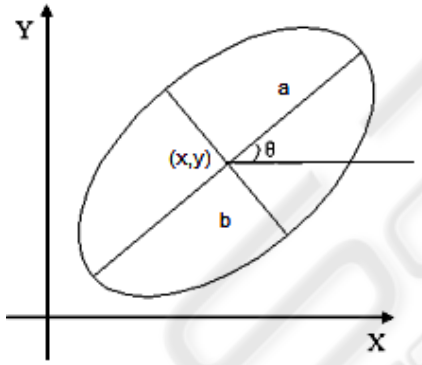


Figure 1: The hand is represented by ellipse model.

Normal boundary box (rectangle model) provides the central position of hand, yet could not properly tell whether a pixel within the box is a hand pixel. As the hand can be considered as an ellipse in most occasions, we use ellipse model to describe a hand for better accuracy. The ellipse model is defined as (x,y,a,b,θ), where (x,y) represents the center of the hand; (a,b) are two axis which indicates the hand size; and θ is the angle between axis a and horizontal axis, which describes the orientation of hand (See Figure 1).

Given a pixel $P(x_p,y_p)$ and a hand ellipse model $E(x_e,y_e,a_e,b_e,\theta_e)$, the relative distance from P to E is defined as D(P,E) which is calculated by following equations.

$$D(P,E) = \sqrt{d_r x^2 + d_r y^2} \quad (9)$$

$$\begin{pmatrix} d_r x \\ d_r y \end{pmatrix} = \begin{pmatrix} \frac{1}{a} & 0 \\ 0 & \frac{1}{b} \end{pmatrix} \left(\begin{pmatrix} \cos\theta_e & -\sin\theta_e \\ \sin\theta_e & \cos\theta_e \end{pmatrix} \left(\begin{pmatrix} x_p \\ y_p \end{pmatrix} - \begin{pmatrix} x_e \\ y_e \end{pmatrix} \right) \right) \quad (10)$$

It is obvious that P is inside model E (including the boundary) when the value of D(P,E) is less than or equals 1 and on the contrary P is outside E when the value is larger than 1. Thus we define the position likelihood function of pixel P and hand model E as

$$p_{position}(P,E) = \begin{cases} 1 & D(P,E) \le 1 \\ D(P,E)^{-2} & 1 < D(P,E) \le T \\ 0 & D(P,E) > T \end{cases} \quad (11)$$

The possibility is set to 100% without any doubt, when the pixel is inside the hand model. And the possibility is set to 0, if the relative distance is larger than certain threshold T (in our system T=16). Otherwise, the possibility is defined by certain expression of the relative distance shown in equation 11.

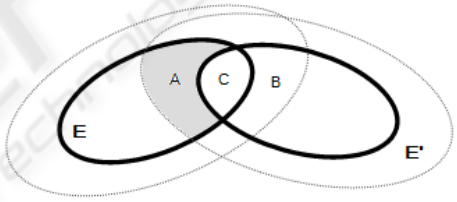


Figure 2: Two ellipse models occlude. For each model, pixels in the black circle are inside the ellipse E and pixels between the black and gray circles probably belong to E.

Considering the contact situation of two models, the value of position likelihood function between P and E will be impacted by model E'. As shown in figure 2, when P is in area B, P is 100% part of E' but is not sure if it is in E. In this situation the possibility $p_{position}(P,E)$ will be reduced by a reduction rate ρ. Usually, ρ is no less than 1, and can even be infinitive in extreme situation. Similarly, when P is in area A, although $p_{position}(P,E)$ remains the original value, the possibility $p_{position}(P,E')$ is reduced. If P is in any other area, both possibilities will stay in original status.

3.3 Motion Information

In the video sequence, two hands will keep moving all the time. This not only makes the hand position change, but also makes the hand size change. In our system, Kalman filters are applied to estimate and

predict the location and also the size of the hand. To model the motion of hand position, we assume that the movement of the hand would be sufficiently small during step interval ΔT. A dynamic process could be used to describe the x and y coordinate at the center of the hand on the image plane with state vectors V (represents either X or Y) which includes the position and velocity of each coordinate. The dynamic process is defined as

$$V_{k+1} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} V_k + \begin{bmatrix} \Delta T^2/2 \\ \Delta T \end{bmatrix} w_k \qquad (12)$$

The system noise is modeled by w_k, an unknown scalar acceleration, whose statistical characteristics are Gaussian and white. Therefore, an observation model can be given by

$$z_{k+1} = HV_{k+1} + v \qquad (13)$$

Where V_{k+1} is the actual state vector at time k+1, v is measurement noise, and z_{k+1} is the central location of the hand model at time k+1.

A similar Kalman filter model is applied to hand size. The length of two axis a and b is also predicted at beginning of each frame.

The motion information is used to select most probably hand status with location result of previous frames.

3.4 Possibility Support Map

Combining these three aspects of information, the PSM with the same size of the frame is generated. For every object, a channel is constructed. The value at position (x,y) is the possibility (denoted as p(P,E)) that pixel $P(x_p,y_p)$ with color $C(x_p,y_p)$ supports the corresponding hand model $E(x_e,y_e,a_e,b_e,\theta_e)$. There are three steps to calculate p(P,E).

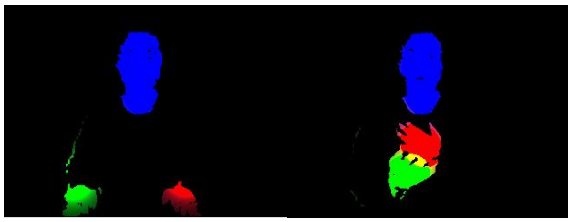
Firstly, we get predicted hand model by motion information as soon as a new image frame is acquired, where the procedure proposed in last subsection is carried out.

The next step is to calculate position likelihood for pixel P. Using equation 11 and considering the reduction effect produced by other hand, $p_{position}(P,E)$ is gained.

The last step is to calculate color likelihood for pixel P. $p_{color}(P)$ is determined by p(skin|C) which is calculated by equation 1.

Finally, support possibility of pixel P for model E is defined as

$$p(P,E) = p_{position}(P,E) \times p_{color}(P) \qquad (14)$$



(a) (b)

Figure 3: These two images show the possibility support map. (a) shows that the brightness of a color indicates the possibility difference of corresponding model. (b) shows that two models overlap.

Figure 3 shows the PSM of three channels. The blue color represents for head channel, red for left hand and green for right hand. The brighter the color is, the higher the possibility is in the corresponding channel. As we can see in Figure 3b, when two hands contact, some pixels are shown in yellow, which means the pixel probably belongs to two hands at the same time.

4 HAND TRACKING ON POSSIBILITY SUPPORT MAP

As soon as the PSM for current frame is generated, a two-step tracking procedure is applied on each channel of PSM to achieve actual hand/head location.

In the first step, we assume that the size and orientation of the ellipse model will not change. Therefore only the position of the ellipse model moves. Based on the assumption, a mean-shift like algorithm is applied, where search window is replaced by our ellipse model.

1) Choose the initial location of the model E.

2) Calculate the center by following equation.

$$M_{ij} = \sum_{D((x,y),E)\le 1} x^i y^j p(x,y) \qquad (15)$$

$$x' = M_{10}/M_{00}; y' = M_{01}/M_{00} \qquad (16)$$

3) Set the center of the ellipse model E at (x',y').

4) Repeat steps 2 and 3 until convergence (x' and y' move less than a preset threshold).

In the second step, we fix the center of ellipse model and try to find best fit size and orientation parameter for the model. A circle window Q is then constructed, which center is at (x_e,y_e) and radius is 10% longer than the length of longer axis of E (namely a). Within this window, proper parameters are calculated by equations 17 and 18.

$$\theta = \frac{1}{2}\arctan\left(\frac{2\mu'_{11}}{\mu'_{20}-\mu'_{02}}\right) \quad (17)$$

$$axis = \frac{\mu'_{20}+\mu'_{02}}{2} \pm \frac{\sqrt{4\mu'^{2}_{11}+(\mu'_{20}-\mu'_{02})}}{2} \quad (18)$$

$$\begin{cases} \mu'_{20} = M_{20}/M_{00} - x'^{2} \\ \mu'_{02} = M_{02}/M_{00} - y'^{2} \\ \mu'_{11} = M_{11}/M_{00} - x'y' \end{cases} \quad (19)$$

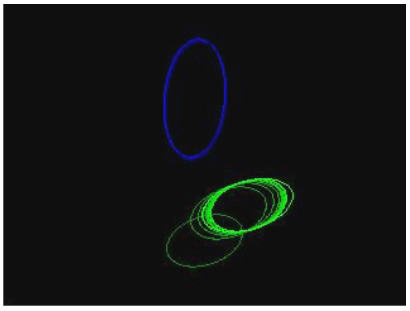


Figure 4: Tracking two objects simultaneously with two-step tracking procedure, which firstly locates position of ellipse and secondly refines size and orientation parameter.

Figure 4 shows the iteration procedure, where blue ellipse stands for head object and green ellipse represents left hand. As left hand moves toward right, it takes 8 iterations to fit the new position. We can see that the new size which is calculated then is larger than the original and the orientation hardly changes.

At the end of locating hands position of current frame, the update process would be carried out. Equations 5~7 are used to update color information and meanwhile correction procedure is applied for Kalman filter with newly located hand status (center position and size).

5 EXPERIMENT RESULT

Our prototype system works at 10~14 frames per second on a laptop with 2.1GB Hz Duo CPU and 2.0GB RAM under 320*240 image resolution. The speed would be faster after code optimization on a better hardware for practical system. It is been tested that with smaller image resolution the fps will be rapidly increased however the accuracy will be decreased significantly.

The user is sitting in front of the screen on which the camera is placed. The working environment is under normal office illumination conditions.

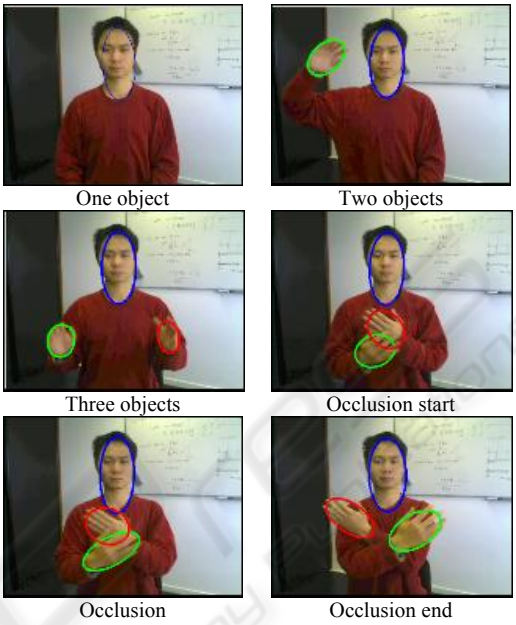


Figure 5: Result of hand locating with and without hand occlusion.

Figure 5 shows the hands tracking result of a video sequence. Head is represented as blue ellipse and two hands are respectively represented as red and green ellipses. We can see that our tracking approach works well when hands are moving in and out of frame.

When one hand is close to the other one (or head), the approach distinguishes the two objects correctly. Even when one hand is covering the other one, we can still obtain efficient result. Figure 6 shows the result of our method and normal camshift tracking algorithm (provided in OpenCV) at the same time.

As the result shown in Figure 6, the Camshift algorithm tracks effectively when hands are isolated respectively, but when the two hands get close, their models are badly influenced by each other (Frame 309) and even collapse into one model (Frame 369). Our method tracks different hand in their own PSM channel. Although these PSM channels are generated from the same video frame, they are carrying different information due to their model parameter and the relationship between themself.

Our proposed method tracks well when two hands overlaps even when two hands are in front of the face at the same time Nevertheless, as some part of the hand is covered by other one, its size while

unavoidably may slightly differ from actual size due to its 'invisibility'.

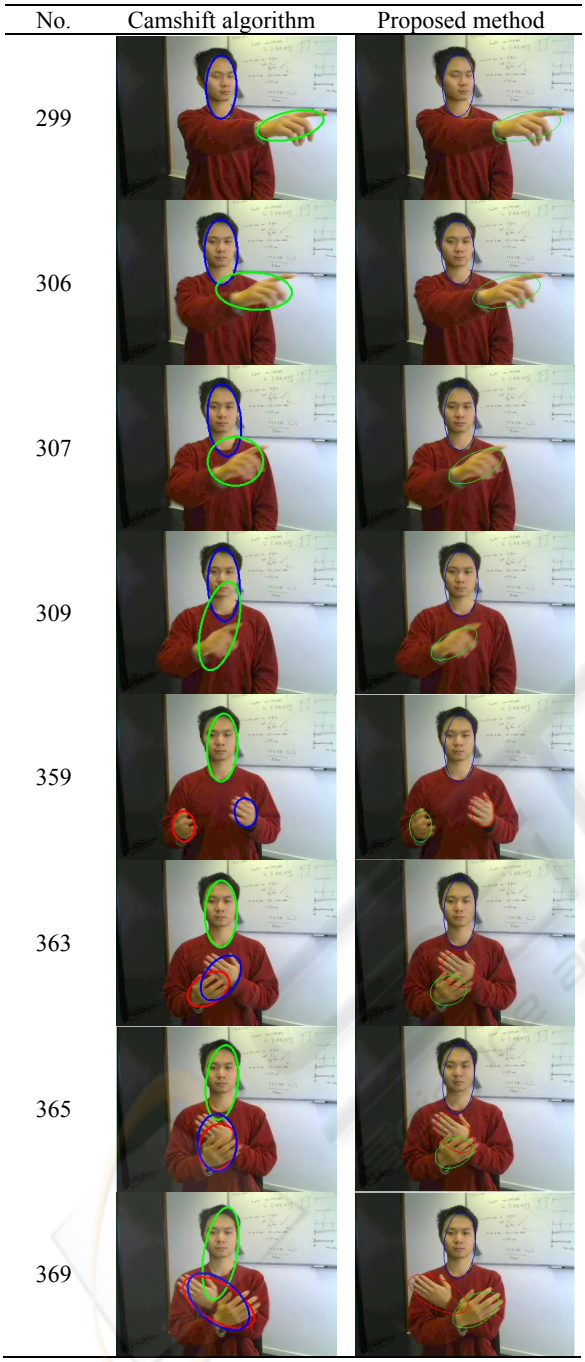


Figure 6: Comparison between locating result of proposed method (on the right column) and tracking result of Camshift algorithm (on the left column). Camshift will track the wrong object or track two object as a whole, while our proposed method can separately locate tracked objects.

The learning rate r mentioned in Equation 5 to 7 will impact on final result when illumination changes. On extreme situation where r equals zero, it means no adaptive update process is carried out: fixed parameters will be used. Thus chaos occurs due to p(C|skin) never changes when light condition changes. On the other hand, r should not be to large. On another extreme situation where r equals 1, it means no previous information is used. Thus the color information process will not be stable, any light noise will lead the update to failure. For normal office illumination condition, r is set to 0.05. And r should be adjust to better value on various illumination conditions.

6 CONCLUSIONS

In this work, we have proposed a robust real-time hands tracking system based on monocular vision by generating possibility support map from hand color information, position information and motion information. For the locating procedure, the system applied a two-stage search algorithm which adapts both the center position change and size change. The current prototype system can process the image up to 10 to 14 frames per second, furthermore the system can provide precise hand status, including position, size and orientation, even if hand occlusion happens. Our next step is to integrate this hand locating method into HCI application which allows user to drive his avatar in remote collaborative virtual environment.

REFERENCES

Tamura, S., Kawasaki, S., 1988. Recognition of Sign-language Motion Images. Pattern Recognition, 21:343-353.

Starner, T., Pentland, A., 1995. Real-time American Sign Language Recognition from Video using Hidden Markov Models. *In Proc. International Symposium on Computer Vision*, pp. 265-270.

Hasanuzzaman, M. et al., 2004. Real-time Vision-based Gesture Recognition for human robot interaction. In *IEEE International Conference on Robotics and Biomimetics*, pp. 413-418.

Han, J. et al., 2006. Automatic Skin Segmentation for Gesture Recognition Combining Region and Support Vector Machine Active Learning. In *International Conference on Automatic Face and Gesture Recognition*, pp. 237-242.

Lee, M. W., Cohen, I., 2004. Human Upper Body Pose Estimation in Static Images. In *European Conference*

on Computer Vision, pp. 126-138.

Askar, S. et al., 2004. Vision-based Skin-colour Segmentation of Moving Hands for Real-time Applications. In *European Conference on Visual Media Production,* pp. 79-85.

Comaniciu, D., Ramesh, V., 2000. Mean Shift and Optimal Prediction for Efficient Object Tracking. In *International Conference on Image Processing*, pp. 70-73.

Bradski, G. R., 1998. Computer Vision Face Tracking for Use in a Perceptual User Interface. *Intel Technology Journal* 2:12-21.

Vacavant, A., Chateau, T., 2005. Realtime Head and Hands Tracking by Monocular Vision. In *IEEE International Conference on Image Processing,* pp. II-302-5.

Schreer, A. et al., 2005. Real-time Avatar Animation Steered by Live Body Motion. In *International Conference on Image Analysis and Processing*, pp. 147-154.

Kirubarajan, T., Bar-Shalom Y., 2001. Combined Segmentation and Tracking of Overlapping Objects with Feedback. In *IEEE Workshop on Multi-Object Tracking*, pp. 77-84.

Coogan, T. et al., 2006. Real Time Hand Gesture Recognition Including Hand Segmentation and Tracking. In *International Symposium on Computer Vision,* pp. 495-504.

Imagawa, K. et al., 1998. Real-time Tracking of Human Hands from a Sign Language Image Sequence. In *Asian Conference on Computer Vision*, pp. 698-705.