Proceedings of the 2012 IEEE
International Conference on Robotics and Biomimetics
December 11-14, 2012, Guangzhou, China

# Robust Hand Tracking with Refined CAMShift Based on Combination of Depth and Image Features

Wenhuan Cui, Wenmin Wang* and Hong Liu

*Abstract*— **Hand tracking is essential for natural Human Robot/Computer Interaction (HRI/HCI), although efficient and robust hand tracking in complex environment is still a challenging issue. While most researchers simplify the issue by strictly controlling the environment with many restrictions on users' clothing, or the scene complexity, or hand motion, this paper focused on reducing these restrictions. As one major cause of the restrictions is the lack of depth information, this paper proposed a method combining depth cues with image features. Depth and motion cues were extracted through background subtraction and histogram based segmentation. Guided by the depth cues extracted, color image features were then extracted with skin-color region segmentation. Then different cues were fused adaptively to construct a probability map for the hand to be tracked. With this map, a refined CAMShift tracking scheme was developed. And based on hand direction constraints we conjectured empirically, a further refinement step was proposed to segment hand from forearm, which is usually avoided using restrictions on clothing for simplicity. A number of experiments were performed to demonstrate the method's effectiveness and robustness. Tracking rates in the experiments are around 85% for ordinary situations, and around 75% for complex situations, such as fast hand motion and distractors.**

## I. INTRODUCTION

Bare hands are probably the most natural HRI/HCI "tools" for its dexterity, and have shown great potential in applications for intelligent electronics, virtual reality, robots and many other fields. Vision based hand tracking, including global hand tracking (localization) and gesture recognition, has been extensively studied for decades. However, fast and robust hand tracking is not yet achieved, as gesture recognition and global hand tracking are still of great challenge in complex scenarios.

In the past decades, plenty of methods have been developed to tackle the problem of hand tracking. *Hand tracking* refers to both global hand tracking and gesture recognition [1][2] [3] [4]. Generally these methods can be categorized into two groups: model-based (or generative) and appearance-based (or discriminative). Model-based approaches try to fit the image observation to the hypothesized model, either an articulated model or a rigid/deformable template. For gesture recognition, Rehg and Kanade in [5] constructed a typical 27-DoF articulated hand model, Ahmad in [6] devised a statistical shape model, and Heap and Hogg in [7] proposed deformable templates. Stenger et al. in [8] used a hierarchical Bayesian filter for both global hand tracking and pose estimation. These model-based methods are more robust but less efficient, due to the high dimensionality of the model space along with the ambiguities in the mapping between model space and image/feature space, constraining their use in laboratories. On the other hand, appearance-based methods are based directly on image features, without the overload of maintaining a sophisticated model. Skin color is a prominent feature for hand and face tracking. CAMShift used mean-shift with color histogram and back projection to track human face in videos [9], which shares some similarity with global hand tracking. Shan et al. in [10] combined particle filter and mean-shift for global hand tracking and performs basic gesture recognition based on orientation histogram. And Mohr et al. proposed a skin color based hand tracking method using multiple cameras [11]. Donoser and Bischof in [12] used the Maximally Stable Extremal Regions (MSER) tracker with color likelihood maps for hand tracking. Contour feature was used with particle filter in [13] to devise a bare-hand drawing interface. The appearance based methods are relatively faster and can achieve remarkble performance, but are susceptible to noise, fast motion and occlusion situations.

Most of the above researches assume restrictions on users' clothing, the scene and hand motion speed, thus limiting the robustness of tracking. Fusing multiple cues is an efficient paradigm for increasing the robustness of object tracking under complex situations, which is also shown in our former works for human tracking [14][15][16]. Recent years, depth cameras are becoming affordable for ordinary users. Depth images are resistent to illumination chage, and provide ample information about the geometry of object surfaces. Reference [19] took advantage of the depth sensor Kinect [17] to build a remarkable system for human pose recognition. For hand tracking, the potential of depth information needs further exploration. One example can be found in [18], which shared much similarity with our work in spirit, but differed in implementation details. In our work, we expected to overcome those restrictions by extracting and combining depth, motion and skin color features without the sacrifice of tracking performance. Each feature was extracted using efficient method separately, to limit the overall complexity. As depth cues were the most stable, adaptive fusion method with emphasis on depth cues were developed. With a confidence map resulting from the fusion, hand tracking with increased robustness was achieved using appearance based methods like CAMShift. An overview of the workflow is

Wenhuan Cui is with the Engineering Lab on Intelligent Perception for Internet of Things(ELIP), Shenzhen Graduate School,Peking University,China josephcuiwh@gmail.com

*Correspongding Author: Wenmin Wang is with the Engineering Lab on Intelligent Perception for Internet of Things(ELIP), Shenzhen Graduate School,Peking University,China wangwm@pkusz.edu.cn

Hong Liu is with the Engineering Lab on Intelligent Perception for Internet of Things(ELIP), Shenzhen Graduate School,Peking University,China hongliu@pku.edu.cn

978-1-4673-2127-3/12/$31.00 © 2012 IEEE

illustrated in Fig. 1.

The remaining content is organized as follows. Section II describes feature extraction methods and the feature integration method for generating the confidence map, which can be used both for hand detection and tracking. Section III describes the modified CAMShift tracking with blob refinement to segment hand from arm, along with a description of our automatic tracking initialization stage. Section IV presents experimental results and discussions. The last section concludes the benifits and limitations of our approach and future directions for improvement.

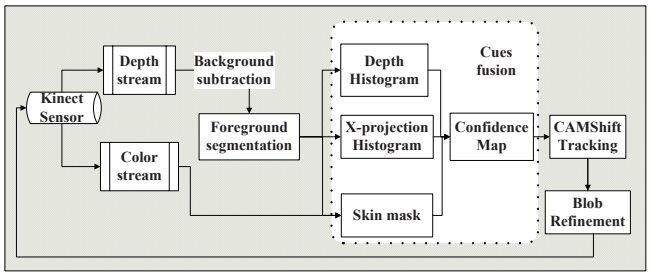



Fig. 1. The work flow of the proposed approach

## II. FEATURE EXTRACTION AND INTEGRATION

### A. Depth image based foreground segmentation

There is extensive literature on motion detection and segmentation based on background modeling [23] [24]. We chose the Codebook model for its simplicity and efficiency. Depth images have the advantage of resistence to illumination change, thus is chosen for background modeling. Since adjacent pixel values are often correlated, we propose to down-sample the image and build codebooks for the down-sampled one, and after background subtraction, up-sample the motion mask image back. Indeed this shows no significant decrease in performance but saves the computation resource.To encode more information in one codeword, we can construct a 6-dimensional vector as in [20] [25]. Use the following 6-dimensional vector to denode a codeword $\mathbf{c}$:

$$\mathbf{c} = (L_{high}, L_{low}, Max, Min, MNRL, freq) \quad (1)$$

where $L_{high}$ and $L_{low}$ are used to determine the tolerance for codeword matching. MNRL, Max Negative Run Length, is the longest time that the codeword has not been visited. Combined with the codeword frequency $freq$, the number of codewords in a codebook can be adaptively updated.

During training, a match in the current codebook for every new pixel value $p$ is defined as:

$$L_{low} \leq p \leq L_{high} \quad (2)$$

As soon as a match is found, the corresponding codeword is updated. Otherwise, a new codeword is added and initialized with $p$:

$$L_{low} = p - bias, L_{high} = p + bias \quad (3)$$

After training, the codebook is used for motion detection. The match criterion in detections stage is:

$$Min - bias \leq p \leq Max + bias \quad (4)$$

The $bias$ is added to increase the robustness of the detection. Typical background subtraction results are shown in Fig.2.

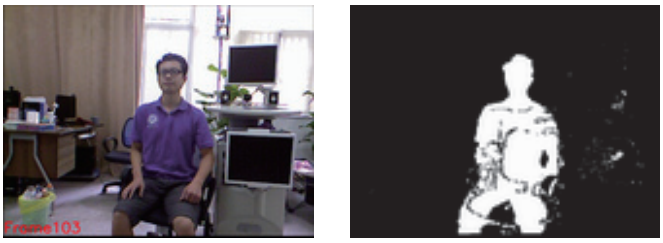


Fig. 2. Foreground segmentation in clutter

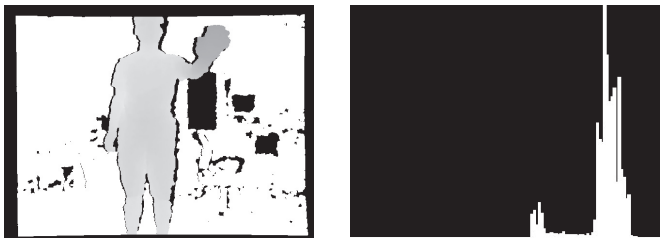
### B. Histogram-based arm/hand segmentation

The refined foreground can also be treated as a mask indicating which pixels are relevant. Thus we can build a depth histogram just for relevant objects in the depth image, instead of for all pixels, to save space and time. If the person in the scene stretches ahead his/her hand for interaction, then depth thresholding can produce a depth mask $\mathcal{D}$, which helps segment the hand, with the threshold chosen by histogram analysis described later. If the person stretches laterally his/her hand, "x-projection" histogram, or $H_x$, can be constructed by projecting the motion mask pixels onto the x (horizontal) axis after a proper binning of the x axis. Denote a pixel as $p = (x, y, z)$, and a function $x(p)$ returns $p$'s x coordinate, $H_x$ for an image $I$ is obtained as follows:

$$H_x(j) = \sum_{i=1}^{|I|} \delta(L_j < x(p_i) < U_j), p_i \in I, j = 1, .., N \quad (5)$$

where $\delta$ is an indicator function, N is the number of bins, and $L_j$ and $U_j$ are the lower and upper boundary value for $j$th bin, respectively. Using $H_x$, the laterally stretched hand can also be marked in a position mask $\mathcal{X}$ produced also by histogram analysis detailed below. One example of these masks is shown in Fig. 3.

The histogram analysis scheme is used both in depth histogram-based and x-projection histogram-based thresholding. First, the maximum of the histogram bin values is found, and then a proportion of this maximum is used as a threshold to threshold the histogram, generating a binary array with 0s and 1s. "Foothills" are found in this array, which are around peaks. And our observation is that "01" transition and "10" transition indicate foothills, and they should occur in pairs. So recording each pair of such transitions gives us foothills pair-wisely. Then for depth histogram, the front-most foothill is chosen as a threshold to segment the depth image so that a hand can be separated from the body. And for the x-projection histogram, the first foothill on the left and on the right can be used as thresholds for the same purpose. The details of this foothill-finding algorithm is shown in TABLE



(a) depth histogram

(b) x-projection histogram

Fig. 3. Depth histogram and x-projection histogram segmentation

Fig. 4. Scaled x-mask

I. Using those foothills we can get a "x-mask", which masks out the body part and leaves out the hands and arms. Next step is to scale its pixel values according to their distance to the body, assigning higher value to pixels farther away from the body trunk, as shown in the right panel of Fig. 4. This step will guide the mode seeking toward the hand in later mean-shift process.

TABLE I

THE FOOTHILL-FINDING ALGORITHM

| The foothill-finding procedure |
|---|
| 1) Find $M = max(H)$ |
| 2) Choose a threshold $s$ such that: $0 < s < 1$. And threshold $H$ with $s$ to get a binary array $H'$, i.e: $$H'(j) = \begin{cases} 1, & H(j) > s \cdot M \\ 0, & H(j) \leq s \cdot M \end{cases}$$ Then a $H'$ is in the form [..011..100..] |
| 3) From left to right, first find a "01" transition, then find a "10" ensuing it, and store the pair, and repeat this process from the right of this "10". Notice boundary situations should be tackled. |

### C. Skin color feature

Skin color has been extensively studied and proven to be useful and robust in face detection, localization and tracking [21]. We chose Peer et al.'s method because it provides robust skin color cues for the hand to be tracked and is fast enough. In this method an RGB pixel is classified as skin if it satisfies the following inequalities [22]:

$$\begin{aligned} &R > 95\, and\, G > 40\, and\, B > 20\, and \\ &max\{R,G,B\} - min\{R,G,B\} > 15\, and \\ &|R - G| > 15\, and\, R > G\, and\, R > B \end{aligned} \tag{6}$$

Notice the last line above as a sufficient condition can be simplified as:

$$R - G > 15\, and\, R > B$$

and thus eliminating the absolute value calculation. The simplicity with regard to the effectiveness of this method is remarkable. However, these inequalities using RGB values are susceptible to illumination variation. Through *histogram equalization* of the Y channel of the corresponding YCrCb space, the result can be improved at the cost of a minimal computation load.This skin segmentation process will give us a skin color mask $\mathcal{S}$.

### D. Integration of features

We explored two ways to integrate the features. The first one is aiming for hand detection, and the second one is for CAMShift-based hand tracking. Hand detection is useful for automatical initialization of the tracking process, and for resuming the tracking after tracking loss. Since in our scenario depth cues are of essential importance, and depth based foreground mask provides a basic mask for image feature extraction, we construct a final mask for hand detection in the following way:

$$\mathcal{F} = \begin{cases} \mathcal{S} \cap \{\mathcal{D} \cup \mathcal{X}\}, & T_l <= |S| <= T_u \\ \mathcal{D} \cup \mathcal{X}, & \text{otherwise} \end{cases} \tag{7}$$

One thing should be noticed is that in the above equations the number of nonzero values in the skin color mask is used to check its reliability. This is based on the assumption that after histogram normalization, skin regions should show similar response, and can be detected or missed simultaneously. In the former circumstance, the skin color mask should be neither too large nor too small, since skin-color regions in the scene usually occupies a proper portion. In the latter circumstance, the skin mask provides no information and thus is discarded. If the skin color mask is reliable, then the final mask $\mathcal{F}$ is the intersection of color mask and the union of depth mask and position mask, otherwise, $\mathcal{F}$ is just the union of depth mask and position mask.

For integration of features, weighted sum is a natural choice. Adjusting the weights is essential but difficult. In this paper the following scheme is used to adaptively change the weights to obtain a better final mask. The final mask is calculated as:

$$\mathcal{M} = w_1\mathcal{S} + w_2\mathcal{D} + w_3\mathcal{X}. \tag{8}$$

And $\mathcal{M}$ will be the probability map used for CAMShift tracking. It is chosen empirically to use the blob size in the depth mask ($s1$) and in the x-mask ($s2$) as a heuristic to

adjust the weights. When this size is below a threshold, we set:

$$w_1 = 0.8, w_2 = w_3 = 0.1 \tag{9}$$

which is to say, let the skin color dominate the mask. Otherwise, we set:

$$w_1 = 0.2, w_2 = 0.8\frac{s1}{s1+s2}, w_3 = 0.8\frac{s2}{s1+s2} \tag{10}$$

The final mask can be thresholded for primary hand detection. Through connected component analysis, hand blob candidates is ranked according to their size and distance to the hand location in the last frame. The blob with the highest rank is chosen to be the detected hand. This detection result will be taken as the tracking result when CAMShift tracking fails.

## III. Tracking with refined CAMShift

CAMShift utilizes mean-shift to locate the local mode of certain distribution. The original CAMShift algorithm is designed for face tracking, using color histogram and backprojection to generate probability map. And after the convergence of the mean-shift process, adaptively update the search window's size and location. The moment based elliptic shape representation used in the original CAMShift algorithm is also useful for hand tracking. However, the ellipse calculated is often the whole forearm. It should be refined to give a reasonable hand tracking result, which begins with the ellipse calculation.

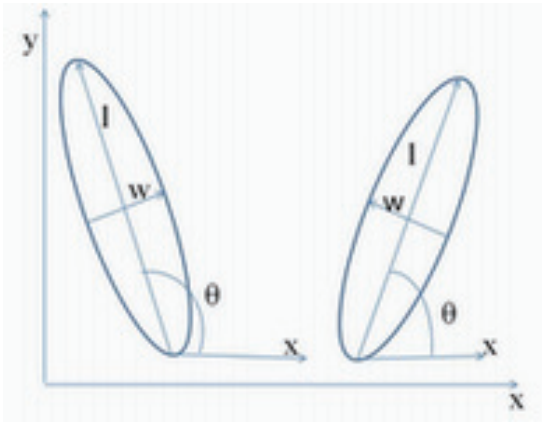
### A. Moment-based elliptic shape representation

Using contour moments we can calculate the ellipse properties: center $\mathbf{p_c} = (x_c, y_c)$, major axis $l$, minor axis $w$, and rotation angle $\theta$, as follows:

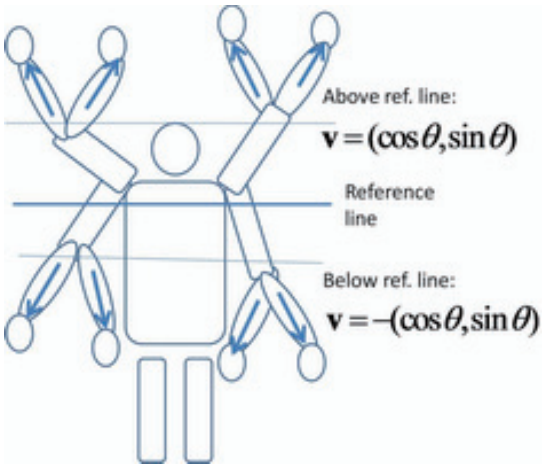
$$\begin{aligned} &m_{00} = \sum_x \sum_y I(x,y). \quad m_{10} = \sum_x \sum_y xI(x,y). \\ &m_{01} = \sum_x \sum_y yI(x,y). \\ &x_c = \frac{m_{10}}{m_{00}}. \quad y_c = \frac{m_{01}}{m_{00}}. \end{aligned} \tag{11}$$

The rotation angle of an ellipse is defined as the angle of the major axis rotated from the positive direction of x axis, and is defined to be in the range of $(0°, 180°)$, as shown in Fig. 5(a):

$$\begin{aligned} &m_{20} = \sum_x \sum_y x^2 I(x,y). \; m_{02} = \sum_x \sum_y y^2 I(x,y). \\ &b = \frac{m_{11}}{m_{00}} - x_c y_c. \quad a = \frac{m_{20}}{m_{00}} - x_c^2. \\ &c = \frac{m_{02}}{m_{00}} - y_c^2. \\ &\theta = \frac{1}{2}\arctan(\frac{2b}{a-c}). \\ &l = \sqrt{\frac{1}{2}((a+c) + \sqrt{4b^2 + (a-c)^2})} \\ &w = \sqrt{\frac{1}{2}((a+c) - \sqrt{4b^2 + (a-c)^2})} \end{aligned} \tag{12}$$




(a) ellipse representation




(b) hand direction constraints

Fig. 5. Elliptic hand representation and hand direction constraints

The aspect ratio [26]:

$$r = \frac{m_{20}}{x_c^2} / \frac{m_{02}}{y_c^2} \tag{13}$$

can be used to update the search window for the next frame:

$$\begin{aligned} \text{width} &= 2\sqrt{m_{00}/256} * r \\ \text{height} &= 2\sqrt{m_{00}/256}/r \end{aligned} \tag{14}$$

### B. Aspect ratio based blob refinement

Many researches on hand tracking circumvent the problem of segmenting hand from forearm, by requiring the subject to wear long sleeves or a wrist band with specific colors. To overcome those requirements, we explored how to estimate the real location of hand in the elliptic blob we had obtained.

Our first observation is that for typical human motion, hand direction is confined differently in different regions surrounding the body. The arrow in Fig. 5(b) denotes the hand direction in the the ellipse representing the forearm. Above a certain reference line $y_{ref}$, forearm is usually upwards (roughly). And below certain line (can be different from $y_{ref}$), hand is usually downwards, although the ellipses look similar. This means the hand direction vector assigned

to the specific ellipse $\mathbf{v}$ is:

$$\mathbf{v} = \begin{cases} (\cos\theta, \sin\theta), & y_c > y_{ref} \\ -(\cos\theta, \sin\theta), & \text{otherwise} \end{cases} \quad (15)$$

Interestingly, the hand direction vector is always $(\cos(\theta), \sin(\theta))$ above the reference line, and $(-\cos(\theta), -\sin(\theta))$ below the reference line, no matter the ellipse is on the right or left of the person, based on our ellipse rotation angle definition.

Our current refining strategy is to move the center toward hand for some amount:

$$\mathbf{p}' = \mathbf{p} + l\mathbf{v}. \quad (16)$$

and reduce the $l$ of the ellipse, untill a proper aspect ratio $l/w$ is obtained. If the reference line is chosen properly, this can produce desired results (Fig. 6)Of course singular points or singular lines exist, especially over the region transition boundary.

Fig. 6. Aspect ratio based blob refinement

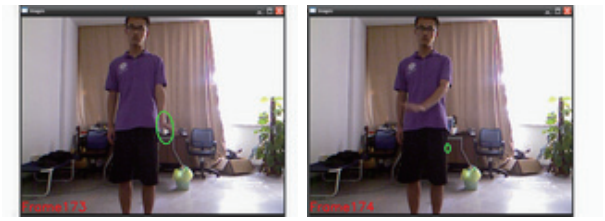
### C. Tracking in difficult situations

The first major difficulty is fast motion, which can be well tackled through re-detection. In fact, the probability map provides a fine basis for hand detection. Hence when tracking fails, mainly caused by the local property of mean-shift, detection can take over the process to resume the tracking. The fast motion problem is well addressed using the tracking-combining-detection approach (Fig. 7).

The other major difficulty is that the face is often a distractor for the hand tracking process. When hand moves fast across face, the tracker can be easily trapped in the face region, due to the local property of mean-shift process. Therefore the size and zero moment of the tracked region is checked, to find out the trapped condition. If the trapping has happened, detection process can take over, thus overcoming local trap (Fig. 8).

## IV. Experiments and Discussions

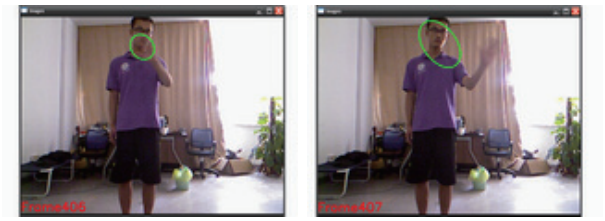
To evaluate the proposed method we conducted a number of experiments. For performance comparison, we compared our method with two well-known methods, the Kernel Particle filter [10] and the original CAMShift tracker. It is observed that the original CAMShift tracking implemented in OpenCV ("standard CAMShift") does not give valid hand tracking result most of the time, as it is based on hue histogram. As shown in Table II, our approach provides much better tracking result, under the same experimental video sequences. The proposed system takes 4.8 seconds for training, which is about 10 FPS. During detection and tracking, the processing is about 7 FPS. Notice that the size of the images in our experiments is four times larger than images in [10], the proposed system indeed is still advantageous with respect to speed. And with blob refinement and fusion of multiple cues, our method outperforms the original CAMShift algorithm in terms of accuracy and robustness.



(a) tracking with only CAMShift

(b) tracking with proposed method

Fig. 7. Tracking fast movement



(a) tracking with only CAMShift

(b) tracking with proposed method

Fig. 8. Tracking with Skin distractor

TABLE II

Comparison of Overall Performances

| Method | Platform | Image size | FPS | Tracking rate |
|---|---|---|---|---|
| KPF[10] | 2.4GH Pentium | 240×180 | 12 FPS | unknown |
| Standard CAMShift | 2.3GH AMD, 3G Memory | 640×480 | 5.8 FPS | ≤50% |
| proposed | | | 6.9 FPS | 70∼89% |

(a) Refined CAMShift with color cue

(b) Multi-cue CAMShift without refinement

(c) The proposed approach

Fig. 9. Comparison of three tracking methods

### A. Tracking performance

The assumption is that camera is still, and user is facing it. The results from different video sequences are shown in Table III. Different difficulties are tested in different video sequences, about 2500 frames in total. Most of the tracking errors happen on wrist or forearm under special body postures. Better results are obtained when we considerd ordinary interaction-oriented hand motion. Noticeably two hand motion and arbitray/drastic hand motion are not resolved quite well currently, causing many tracking errors.

TABLE III

VIDEO DESCRIPTION AND EXPERIMENTAL RESULTS

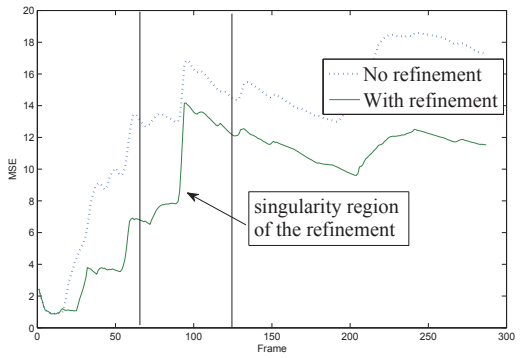
| Seq. NO. | Characteristics | Tracking Frames | Correct Tracking | Tracking rate |
|---|---|---|---|---|
| Seq.1 | large motion | 282 | 215 | 76.2% |
| Seq.2 | Day,Normal | 314 | 274 | 87.3% |
| Seq.3 | Night,Normal | 251 | 225 | 89.6% |
| Seq.4 | two hand motion | 369 | 266 | 72.1% |
| Seq.5 | Sitting person | 337 | 202 | 89.6% |
| Seq.6 | Not always facing camera | 418 | 328 | 78.5% |
| Seq.7 | fast motion, occlusion | 552 | 434 | 78.8% |

We compare three tracking methods: standard CAMShift, multi-cue CAMShift without refinement, and the proposed method. As shown in Fig.9, ellipse denotes the tracked hand. Color cue is unstable, especially when the illumination condition is not ideal (Fig.9(a)). With multiple cues, especially combining depth cues with image features, plus proper weighting, better tracking results can be achieved (Fig.9(b)). Also can be seen is the long thin ellipse without refinement. Using refinement hand tracking output with higher accuracy can be achieved (Fig.9(c)).

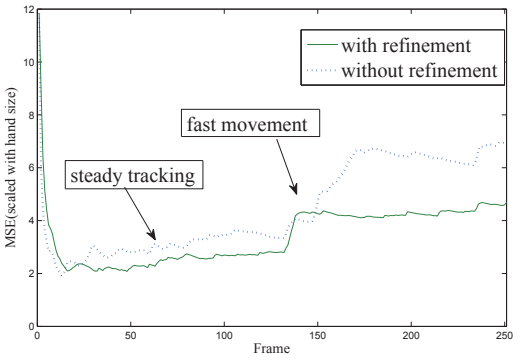
Lastly the Mean Squared Error (MSE) is used as a quantified assessment of tracking performance. As shown in Fig. 10, the proposed method can get smaller MSE, and thus improved tracking accuracy.

## V. CONCLUSIONS AND FUTURE DIRECTION

This paper studies the problem of tracking global hand motion for HCI/HRI, focusing on reducing the restrictions applied by common hand tracking researches. Without depth information 2D hand tracking can hardly be obtained except under this restrictions, and is not robust under complex situations. By combining depth cues, motion cues and color cues extracted from depth and color images, our method demonstrates greatly improved robustness, especially under fast motion and skin-color distractor situations. For cues extraction, various techniques are proposed to increase time efficiency. The down-sampled Codebook background model reduces the computational complexity of foreground segmentation. Histogram calculation with a foothill-finding scheme is shown to be efficient and reliable for arm/hand segmentation. The adaptive fusion of multiple cues increases the robustness of CAMShift tracking. The aspect ratio-based blob refinement scheme proposed for segmenting hand from arm, provides hand locations with higher accuracy. Experiments have demonstrated the effectiveness and robustness of the proposed approach under ordinary and certain complex situations, such as fast motion and skin-color distractors. However, for arbitrary camera pose, two hand motion or




(a) MSE of tracking on Sequence 1




(b) MSE of tracking on Sequence 3

Fig. 10. MSE of Tracking with and without refinement

arbitrary arm/hand motion, CAMShift has limited tracking accuracy even with depth information. More robust tracking schemes should be developed to tackle these problems in the future.

## VI. ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China(NSFC, No.60875050, 60675025), National High Technology Research and Development Program of China(863 Program, No.2006AA04Z247), Scientific and Technical Innovation Commission of Shenzhen Municipality (No.JC201005280682A, CXC201104210010A).

## REFERENCES

[1] R. Y. Wang and J. Popovic, "Real-Time Hand-Tracking with a Color Glove", *ACM Transaction on Graphics*, SIGGRAPH 2009, 28(3), pp. 63:1-63:8

[2] F. Mahmoudi and M. Parviz, "Visual Hand Tracking Algorithms", In: *Proceedings of the Geometric Modeling and Imaging-New Trends*, London, UK, 2006, pp. 228-232

[3] S. Mitra and T. Acharya, "Gesture recognition: A survey", *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, 37, 2007, pp. 311-324

[4] A. Erol, G. Bebis, M. Nicolescu, R.D. Boyle, X. Twombly, "Vision-based hand pose estimation: A review", *Computer Vision and Image Understanding , Special Issue on Vision for Human-Computer Interaction*, 108(1-2), 2007, pp. 52-73

[5] J. Rehg, T. Kanade, "Digiteyes: vision-based hand tracking for human-computer interaction", in:*Workshop on Motion of Non-Rigid and Articulated Bodies*, 1994, pp. 16-24.

[6] S. Ahmad, "A usable real-time 3d hand tracker", In*Proceedings 28th Asilomar Conference on Signals, Systems and Computers*, IEEE Computer Society Press, 1995, pp. 1257-1261

[7] T. Heap and D. Hogg, "Towards 3d hand tracking using a deformable model", In *2th Conference on Face and Gesture Recognition*, IEEE Computer Society Press, 1996, pp. 140-145

[8] B. Stenger , A. Thayananthan , P. H. S. Torr , R. Cipolla, "Model-Based Hand Tracking Using a Hierarchical Bayesian Filter", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9), 2006, pp. 1372-1384

[9] G. Bradski, "Computer Video Face Tracking for Use in Perceptual User Interface", *Intel Technical Journal*, 1998, pp. 705-740

[10] C. Shan, Y. Wei, T. Tan, F. Ojardias, "Real Time Hand Tracking by Combining Particle Filtering and Mean Shift", In: *International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 669-674

[11] D. Mohr, G. Zachmann, "Real-Time Hand Tracking for Natural and Direct Interaction", in *Whole Body Interaction 2010, A SIGCHI 2010 Workshop*, Atlanta, USA, 2010, pp. 1-6

[12] M. Donoser and H. Bischof, "Real Time Appearance Based Hand Tracking", In *Proceedings of International Conference on Pattern Recognition* (ICPR), 2008, pp. 1-4

[13] M. Isard and A. Blake, "CONDENSATION-conditional density propagation for visual tracking", *International Journal of Computer Vision*, 29(1), 1998, pp. 5-28

[14] H. Liu, Z. Yu, H. Zha, Y. Zou, "Robust Human Tracking Based on Multi-Cue Integration and Mean Shift", *Pattern Recognition Letters*, 30(9), 2009, pp. 827-837

[15] H. Liu, H. He, "A Salient Feature and Scene Semantics based Attention Model for Human Tracking on Mobile Robot", In: *Proceedings of 2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4545-4552

[16] Y. Shi, H. Liu, Y. Liu, H. Zha, "Adaptive Feature-Spatial Representation for Mean-Shift Tracker", In *15th IEEE International Conference on Image Processing*, 2008, pp. 2012-2015

[17] Microsoft Corp. Redmond WA. Kinect for Xbox 360.

[18] A. Bleiweiss and M. Werman. Fusing time-offlight depth and color for real-time segmentation and tracking. In *Dyn3D Proceedings of the DAGM Workshop on Dynamic 3D Imaging*, 2009, pp. 58-69

[19] J. Shotton, A. Fitzgibbon, M. Cook, and A. Blake, "Real-time human pose recognition in parts from single depth images", in *Proc. Conf. Computer Vision and Pattern Recognition*, 2011, pp. 1297-1304

[20] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model", *Real-Time Imaging 11*, 2005, pp. 167-256

[21] V. Vezhnevets, V. Sazonov, A. Andreeva, "A survey on pixel-based skin color detection techniques", *GRAPHICON03*, 2003, pp. 85-92

[22] J. Kovac, P. Peer, F. Solina, "Human skin colour clustering for face detection", in: B. Zajc (Ed.) *EUROCON 2003-Internat. Conf. on Computer as a Tool*, Ljubljana, Slovenia, 2, 2003, pp. 144-148

[23] S. Elhabian, K. El-Sayed, S. Ahmed, "Moving object detection in spatial domain using background removal techniques", *Recent Patents on Computer Science 1*, 2008, pp. 32-54

[24] M. Piccardi, "Background subtraction techniques: A review", In: *Proceedings of International Conference of System, Man and Cybernetics*, 4, 2004, pp. 3099-3104

[25] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly, Cambridge, MA, 2008.

[26] D. Exner, E. Bruns, D. Kurz, A. Grundhofer, and O. Bimber, "Fast and robust CAMShift tracking", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 9-16