# A Real-time Multi-cue Hand Tracking Algorithm Based on Computer Vision

Zhigeng Pan[1,3], Yang Li[1], Mingmin Zhang[1]*, Chao Sun[1], Kangde Guo[1], Xing Tang[2], Steven Zhiying Zhou[4]
[1]State Key Lab of CAD&CG, Zhejiang University
[2]Intel Asia-Pacific Research & Development Ltd
[3]State Key Lab of Virtual Reality, Beijing University of Aeronautics & Astronautics
[4]Department of Electrical Computer Engineering, National University of Singapore

## Abstract

Although hand tracking algorithm has been widely used in virtual reality and HCI system, it is still a challenging problem in vision-based research area. Due to the robustness and real-time requirements in VR applications, most hand tracking algorithms require special device to achieve satisfactory results. In this paper, we propose an easy-to-use and inexpensive approach to track the hands accurately with a single normal webcam. Outstretched hand is detected by contour & curvature based detection techniques to initialize the tracking region. Robust multi-cue hand tracking is then achieved by velocity-weighted features and color cue. Experiments show that the proposed multi-cue hand tracking approach achieves continuous real-time results even for the situation of cluttered background. The approach fulfills the speed and accuracy requirements of frontal-view vision-based human computer interactions.

**Keywords:** Hand tracking, 3D Interaction, Gesture detection.

**Index Terms:** H.5.2 [Information Interfaces and Presentation]: User Interfaces—Theory and methods; I.4.8 [Image Processing and Computer Vision]: Scene Analysis —Tracking;

## 1 Introduction

Being probably the most natural HCI method, using the bare human hand to interact with computers has attracted more and more attention in recent years. Compared to matured detection and tracking techniques for rigid objects [1][2], tracking the bare human hand is very challenging since it is a highly articulated flexible object. Currently, in addition to the popular color lookup table approach, background subtraction and adaptive background models are also widely used [3] for hand tracking. However, these approaches assume that either the background is simple or the camera remains stationary, which do not stand for usual scenarios. Kolsch proposes an idea of tracking flock of features in [4] to locate the hand through a head-mounted camera. This method assumes that the hand region dimension is fixed and not many other objects have the similar color with the skin in the background. Thus it's not suitable for frontal-view HCI.

Incorporating the flock of features and the velocity model, a multi-cue hand tracking algorithm based on velocity weighted features and color cues is presented in this paper to track the hand in real-time. The proposed approach is robust under cluttered background and partial occlusion. Meanwhile, contour & curvature based outstretched hand detection approach of [5] [6] is used to initialize and resize the tracking region automatically.

The outline of the paper is as follows: Section 2 describes the related work and gives a brief introduction of our work; Section 3 presents the adaptive skin object segmentation and the contour & curvature based outstretched hand detection algorithm; Section 4 proposes a multi-cue hand tracking algorithm based on velocity weighted features and color cues; Section 5 presents the experimental results for different video sequences and an application for plane driving simulation; Finally, Section 6 draws some conclusions and gives an outlook on future work.

## 2 Related Work

In general, there are two types of hand tracking approaches: model-based and appearance-based [3] [7]. Model-based approaches [8][9][10] estimate the current hand state by matching a 3D hand model to the observed image features. Such approaches can achieve some good results, however they search the hand in a high dimensional space hence may not be suitable for real-time application. Wang [11] employs a nearest-neighbor approach to track a hand wearing a color glove that is imprinted with a custom pattern. Although this approach reduces the searching dimension, the use of color glove does look natural. Appearance-based approaches can also categorized as: color model based; fiducial marker based; background model based and contour & curvature based. However these approaches work under special requirement of known background, stationary camera or fiducial markers.

Contour & curvature based approach proposed by Lee [5] and Malik [12] for desktop environment is still sensitive to skin colored objects appearing in background. Lu [13] uses Bayesian network (BN)-based multi-cue fusion algorithm for fist tracking without gesture changing. Two downward-pointing cameras are used in [12] to implement a *Visual Touchpad.* Schlattmann [14] uses visual hull approach (three cameras) to track the bare-hands in real-time for 3D games. However all the above approaches use more than one camera and assume simple background, which are not suitable for general scenarios.

In this paper, we propose a multi-cue hand tracking algorithm based on velocity weighted features and color cues. The approach not only meets the three requirements mentioned by Wang Xiying et al. [15], but also gives robust results in cluttered background. The proposed approach includes the following three parts:

(1) Initialization: Initializing the tracking region by means of contour & curvature tracking. Though contour & curvature based approach is not robust enough for hand tracking under cluttered background, the contour & curvature based outstretched hand detection approach can meet the requirements of automatic system initialization.

(2) Tracking: We propose multi-cue hand tracking method based on velocity weighted features and color cues. This approach can track the hand in cluttered background with a single frontal-view camera robustly.

(3) Handling failure: When the tracking fails, the tracking region can be reinitialized to resume the tracking with the help of outstretched hand detection.

## 3 Adaptive Skin Object Segmentation and Outstretched Hand Detection

An off-trained Bayesian model on popular *YCbCr* color space [16], together with an online adaptive segmentation approach, is proposed to detect potential skin regions. Adaptive segmentation makes use of hand regions detected in most recent frames to

*Correspondence author: Mingmin Zhang
Email: javazhangmm@163.com

IEEE Virtual Reality 2010
20 - 24 March, Waltham, Massachusetts, USA
978-1-4244-6238-4/10/$26.00 ©2010 IEEE

detect skin color region in current frame. Subsequently, an improved contour & curvature based algorithm is used to detect the outstretched hand.

### 3.1 Adaptive skin object segmentation

A few snapshots of various hands having different range of skin-tones and poses under varying illumination conditions are taken for training. Segmentation of the object with skin color is done manually by an image editing program. Then each captured image is transferred to a binary mask where white pixels represent skin areas and black pixels represent non-skin areas. The hand images and binary masks are then used to train the skin-color histogram $H_s$ and non-skin-color histogram $H_n$. Given these histograms, the probability of a given *CrCb* vector can be computed as follows:

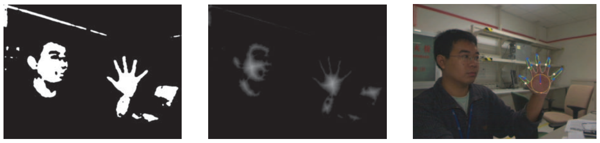
$$\begin{gathered} P_{off}(skin \mid CrCb) = \frac{P(CrCb \mid skin)P(skin)}{P(CrCb \mid skin)P(skin) + P(CrCb \mid \neg skin)P(\neg skin)} \\ P(CrCb \mid skin) = \frac{s[CrCb]}{C_s}, P(CrCb \mid \neg skin) = \frac{n[CrCb]}{C_n} \\ P(skin) = \frac{C_s}{C_s + C_n}, P(\neg skin) = \frac{C_n}{C_s + C_n} \end{gathered} \tag{1}$$

where $s[CrCb]$ and $n[CrCb]$ give the values of the histogram bin in $H_s$, $H_n$ respectively, corresponding to color vector *CrCb*. $C_s$, $C_n$ represent the total counts contained in $H_s$ and $H_n$ respectively. Color vector *CrCb* with probability $P_{off}(skin \mid CrCb)$ larger than a threshold (0.6) are put into the skin color look-up table. In the adaptive segmentation section, the offline and online probabilities are mixed together by Equation (2):

$$P(skin \mid CrCb) = wP_{offline}(skin \mid CrCb) + (1-w)P_{online}(skin \mid CrCb) \tag{2}$$

where *w* is a sensitivity parameter that controls the influence of online and offline training set in the segmentation process.

### 3.2 Outstretched hand detection



(a)Color segmentation (b) Distance transform (c) Hand region detection

Figure 1. Outstretched hand detection

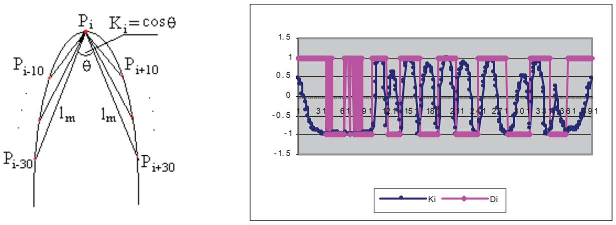
To meet speed and accuracy requirements, contour & curvature based [5] outstretched hand detection is employed to initialize the tracking region. After adaptive color segmenting, a series of skin color regions is obtained (Finger 1(a)). These regions, sorted by their area size $\Omega$, with top 10 and $\Omega$>400 will be the candidate hand regions.

The contour of the region is converted into a series of clockwise vertex points $P_1, \ldots, P_n$. Every point $P_i$ has two attributes, the max cosine value $K_i$ and the direction flag $D_i$:

$$\begin{aligned} K_i &= \max_{10<l<30} K_i(l) \\ D_i &= sign(\overrightarrow{P_iP_{i-l_m}} \times \overrightarrow{P_iP_{i+l_m}}) \\ K_i(l) &= \frac{\overrightarrow{P_iP_{i-l}} \cdot \overrightarrow{P_iP_{i+l}}}{|\overrightarrow{P_iP_{i-l}}| |\overrightarrow{P_iP_{i+l}}|} \\ l_m &= \arg\max_{l \in [10,30]} (K_i(l)) \end{aligned} \tag{3}$$

lower and upper bounds 10 and 30 are set empirically (Figure 2(a)). A point, with curvature value $K_i \geq$ *costhreshold* and direction flag $D_i = 1$, is a candidate fingertip, where *costhreshold* =0.5. Now a series of candidate fingertips is detected. We define two candidate fingertips ($P_i$,$P_j$) as a continuous block where |j-i| < *gap_threshold*. If ($P_i$,$P_j$) and ($P_j$,$P_k$) are continuous blocks, ($P_i$,$P_j$,$P_k$) is combined by ($P_i$,$P_j$) and ($P_j$,$P_k$). The size of the continuous block is the number of candidate fingertips in it. Finger region is the continuous block whose size is bigger than *region_threshold*.

Figure 2(b) shows the max cosine value $K_i$, and the direction flag $D_i$ of every point $P_i$. Five finger regions and four valleys between finger regions ( $D_i = -1$ ) are detected. Distance transformation is performed to find the palm center (Figure 1(b)), and ellipse-fit method is used to detect the fingertips (Figure 1(c)).



(a) Calculation of *Ki* (b) The value of two attributes of each point

Figure 2. Fingertip detection

## 4 MULTI-CUE HAND TRACKING

### 4.1 Initializing the tracking region

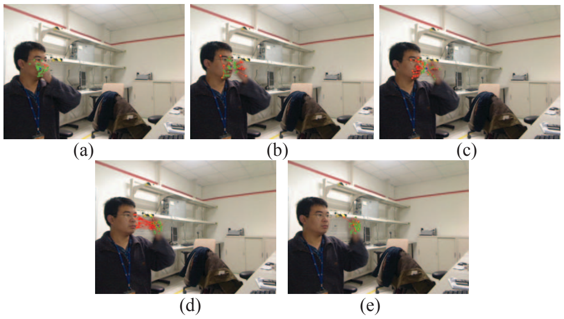
Current hand tracking approaches [4][5][12] usually perform manual selection, fixed region selection or biggest skin color region to initialize the hand region. In this paper, the outstretched hand detection approach, which is presented to initialize the tracking region, makes the interaction more independent, flexible and adaptive.

### 4.2 Weighted feature and color cue based multi-cue hand tracking

Due to the assumption that the hand region size is fixed and there is no other big skin color object in the background, the approach of feature flock [4] is not robust enough. To solve the problems mentioned above, the velocity weighted features are proposed to solve the occlusion problem of other skin color objects. Our approach for flexible hand tracking includes three parts as follows: skin color feature initialization, tracking and relocation, supplement and refinement. Every probable skin color feature should meet the following two requirements: easy to track ("Good Features to Track" [17]) and similar to skin color (Bayesian probability bigger than a threshold (0.6)).

(1) Initializing the skin color features

The speed of pyramid-based KLT feature tracking, which is popular for hand tracking [17], allows breaking through the computation limitation of model-based approaches, achieving real-time performance required by vision based interfaces. After the tracking region is initialized, 100 "Good Features to Track" [19] are selected and ranked by Bayesian color model. Then the top 30 features are selected as the strong skin color features to track.



(a), (b) A lot of features move to the head region while hand moving
(c), (d) Since the features on the hand have high velocity, the features on the head are pulled to the hand region
(e) At last, all features move to the hand region.

Figure 3. Hand and head overlapped

(2) Tracking and relocating the skin color features

The median point $C$ of the flock of features is defined as:

$$C = \arg\min_{i} \sum_{j} \hat{d}_{i,j} \tag{4}$$

where $\hat{d}_{i,j}$ is the velocity weighted distance from skin color feature $P_i$ to $P_j$, as defined in Eq. (5). To ensure the robustness and accuracy of the hand tracking, the features too far away from the center point $C$ or too close to the other features are to be relocated. To prevent interfering with other skin color objects (Figure 3(a)), the velocity weighted distance makes use of not only the pixel distance but also the velocity:

$$\hat{d}_{i,j} = f(\hat{v}_i, \hat{v}_j) * d_{i,j} \tag{5}$$

where $d_{i,j}$ is the pixel distance between $P_i$ and $P_j$, $f(\hat{v}_i, \hat{v}_j)$ is the velocity weighted factor from $P_i$ to $P_j$. The velocity weighted factor $f(\hat{v}_i, \hat{v}_j)$ will be discussed in Section 5.

The proposed algorithm includes three parts as follows:

I. To calculate the median point of the flock of skin color features. With the help of velocity, a weighted digraph is constructed. $C$ in Eq.(4) is the median point of the weighted digraph rather than the pixel center of the flock of features.
II. To remove the singular points. The farthest 10% features from the median point $C$ will be removed to achieve temporally more stable results.
III. To relocate the feature points. To ensure that the features are not too far away from the center point $C$, the feature which is too far away from $C$ will be relocated. The new feature will be the middle point between $C$ and the old position.

(3) Skin color feature supplement and refinement

To maintain continuous hand tracking, we apply fixed number of skin color features. New skin color features will be automatically appended while some old features are lost. New skin color feature will be appended randomly as long as it meets the two requirements: first its Bayesian probability is bigger than threshold (0.6) and second it is at least of 3-pixel distance away from other feature points.

### 4.3 Handling failure

One of the most important issues of hand tracking for HCI is auto-initialization. If tracking fails, the system must be able to re-initialize itself for tracking again without external interfere. In this paper outstretched hand detection is used to resume the tracking automatically when the tracking fails.

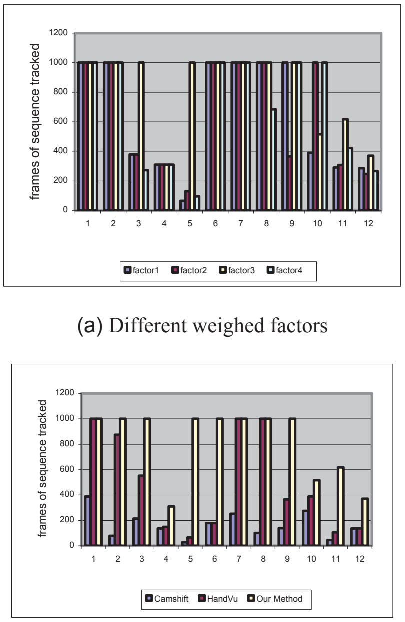
## 5 Experimental results and analysis

Twelve video sequences, catalogued by gesture changes, non-stationary background, indoor/outdoor environment, hand motion changes, and background complexity, are recorded to test the efficiency of the hand tracking algorithms. When recording starts, the hand gesture is an outstretched hand in the camera view. Then the hand moves with the gesture changing, motion speed changing, meanwhile getting occluded with other similar color objects (e.g. the head). A total of 1000 frames video are captured. The detailed information of the video sequences is shown in Table 1.

Table 1 Test video sequences

| Video ID | Gesture change | Background change | Indoor/ Outdoor | The speed of hand motion | Background complexity | Total frames |
|---|---|---|---|---|---|---|
| 1 | N | N | Indoor | slow | few skin color object (single people） | 1000 |
| 2 | Y | Y | Indoor | slow | few skin color object（ single people） | 1000 |
| 3 | Y | Y | Indoor | slow | normal（ single people 、clothes 、wooden door） | 1000 |
| 4 | Y | Y | Indoor | fast | normal（ single people 、clothes 、wooden door） | 1000 |
| 5 | Y | Y | Indoor | normal | complicated（ two people 、wooden furniture） | 1000 |
| 6 | N | N | Outdoor | normal | few skin color object（ single people） | 1000 |
| 7 | Y | Y | Outdoor | normal | normal（ single people 、clothes 、wooden door） | 1000 |
| 8 | Y | Y | Outdoor | normal | normal（ two people 、clothes 、wooden door） | 1000 |
| 9 | Y | Y | Outdoor | normal | normal（ two people 、clothes 、wooden door） | 1000 |
| 10 | Y | Y | Outdoor | normal | complicated（ many people 、street） | 1000 |
| 11 | Y | Y | Outdoor | fast | complicated（ many people 、street） | 1000 |
| 12 | Y | Y | Outdoor | fast | complicated（ many people 、street） | 1000 |

It is very important to select the appropriate velocity weighted factor $f(\hat{v}_i, \hat{v}_j)$. In this paper, following four velocity weighted factors: $\sqrt{\frac{\hat{v}_j}{\hat{v}_i + \hat{v}_j}}$ (Factor1), $\frac{\hat{v}_j}{\hat{v}_i + \hat{v}_j}$ (Factor2), $(\frac{\hat{v}_j}{\hat{v}_i + \hat{v}_j})^2$ (Factor3), $(\frac{\hat{v}_j}{\hat{v}_i + \hat{v}_j})^3$ (Factor4) are tested as candidates. In the video test section (Figure 4(a)), Factor3 achieves the best performance. The velocity with either Factor1 or Factor2 has less influence on the whole flock of features, whereas the center point $C$ with Factor4 is too sensitive to high speed features, which causes an unstable system.



(a) Different weighed factors

(b) Comparing with other methods

Figure 4. Test results of different weighed factors and methods

Our algorithm is better than Camshift and Handvu as shown in Figure 4(b). Camshift is a popular single-cue approach, while Handvu is based on the non-weighted flock of features. The experimental results show that the Camshift is sensitive to the rapid gesture changing and occlusion, thus it can not track the high articulated hand very well. Under the condition of few color objects (video 1, 2, 7, 8), Handvu and our approach both get good results (our approach is better). When the number of skin color objects increases in the scene, our approach outperforms Handvu's approach.

Our proposed approach, far better than the other two algorithms, can successfully track the hand under varying conditions such as complex background, high dynamic hand motion etc.

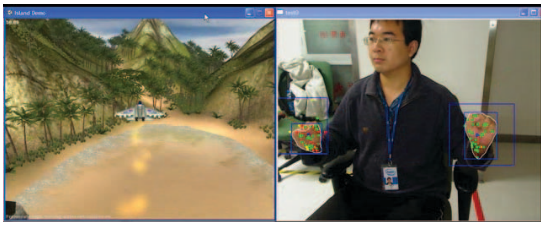


Figure 5. Plane driving

Based on the proposed robust hand tracking approach, a plane driving simulation game has been developed (Figure 5). At the beginning of the game, two outstretched hand are detected by our detection algorithm. After initialization, the motion and position of the two hands are tracked to control the plane to turn left, turn right, climb or dive. Experimental results show that the hand detection and tracking approach is robust in real time.

## 6 Conclusion

In this paper, a robust real-time hand tracking approach is presented by means of the velocity weighted features, combined with the adaptive skin color segmentation. To increase the robustness, multi-cue based approach is used for hand tracking. The moving hand with dynamic gesture can be robustly tracked even in presence of cluttered background. Experimental results show that the successful tracking rate is several times better than Camshift and Handvu. In fact, the successful tracking rate can meet the demand of human computer interaction with single frontal-view camera.

## Acknowledgements

This work is supported by the Intel-University Cooperation Project, NSF of China (No. 60533080), National 863 Program of China (2006AA01Z303) and National 863 Program of China (No. 2009AA062704).

## References

[1]Robert T. Collins. Mean-shift Blob Tracking through Scale Space. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03), Madison, pages 234-240, 2003.

[2] Isard M, Blake A. A mixed-state CONDENSATION tracker with automatic model-switching. Proceedings of the Sixth International Conference on Computer Vision, Bombay, pages 107–112, 1998.

[3] Erol A , Bebis G. , Nicolescu M,Boyle, R.D. ,Twombly, X. Vision-based hand pose estimation: A review, Computer Vision and Image Understanding,108(1-2):52-73, 2007.

[4] Kolsch, M. and Turk, M. Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration. Computer Vision and Pattern Recognition Workshop, Washington, pages 158-158, 2004.

[5] Lee, T. and Hollerer, T. Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking. IEEE International Symposium on Wearable Computers, Boston, pages 83-90, 2007.

[6] Argyros, A.A. and Lourakis, M.I.A. Vision-based Interpretation of Hand Gestures for Remote Control of a Computer Mouse. Lecture Notes in Computer Science, 3979 :40-51, 2006.

[7] Michael Donoser and Horst Bischof. Real Time Appearance Based Hand Tracking. In Proceedings of International Conference on Pattern Recognition (ICPR), 2008

[8] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla.Model-based hand tracking using a hierarchical bayesian filter. IEEE transactions on pattern analysis and machine intelligence, 28(9):1372–1384, 2006.

[9] Sudderth, E. and Mandel, M. and Freeman, W. and Willsky, A. Distributed occlusion reasoning for tracking with nonparametric belief propagation, Advances in Neural Information Processing Systems,17: 1369–1376,2004.

[10] Dewaele, G. and Devernay, F. and Horaud, R. and Forbes, F. The alignment between 3-d data and articulated shapes with bending surfaces. Proceedings of the 9th European Conference on Computer Vision, Graz, pages 578–591, 2006.

[11] Robert Y. Wang and Jovan Popovic. Real-Time Hand-Tracking with a Color Glove, ACM Transaction on Graphics, 28(3), 2009.

[12] Malik, S. and Laszlo, J. Visual Touchpad: A Two-handed Gestural Input Device, Proceedings of the 6th international conference on Multimodal interfaces, State College, PA, pages 289-296, 2004.

[13] Peng Lu,Yufeng Chen,Xiangyong Zeng,Yangsheng Wang. A vision based game control method, Proceedings of the tenth International Conference on Computer Vision, Beijing, pages 70-78, 2005.

[14] Markus Schlattmann, Tanin Na Nakorn, and Reinhard Klein. 3D Interaction Techniques for 6 DOF Markerless Hand-Tracking. In proceedings of International Conference on Computer Graphics, Visualization and Computer Vision (WSCG '09), Feb. 2009.

[15] Xiying Wang, Xiwen Zhang and Guozhong Dai. Tracking of Deformable Human Hand in Real Time as Continuous Input for Gesture-based Interaction, 2007 International Conference on Intelligent User Interfaces (IUI 2007), Hawaii, USA.

[16] Vezhnevets V., Sazonov V., Andreeva A., A Survey on Pixel-Based Skin Color Detection Techniques. Proc. Graphicon-2003, pp. 85-92, Moscow, Russia, September 2003.

[17] Jianbo Shi and Carlo Tomasi. Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition, Seattle, pages 593-600, 1994.