# Hand Pointing Estimation for Human Computer Interaction Based on Two Orthogonal-Views

Kaoning Hu, Shaun Canavan, and Lijun Yin
*Department of Computer Science, State University of New York at Binghamton*

## Abstract

*Hand pointing has been an intuitive gesture for human interaction with computers. Big challenges are still posted for accurate estimation of finger pointing direction in a 3D space. In this paper, we present a novel hand pointing estimation system based on two regular cameras, which includes hand region detection, hand finger estimation, two views' feature detection, and 3D pointing direction estimation. Based on the idea of binary pattern face detector [15], we extend the work to hand detection, in which a polar coordinate system is proposed to represent the hand region, and achieved a good result in terms of the robustness to hand orientation variation. To estimate the pointing direction, we applied an AAM based approach to detect and track 14 feature points along the hand contour from a top view and a side view. Combining two views of the hand features, the 3D pointing direction is estimated. The experiments have demonstrated the feasibility of the system.*

## 1. Introduction

Hand gesture is an efficient means for humans interacting with computers [9, 13, 17]. The most basic and simplest gesture is pointing. Pointing gesture can resolve ambiguities derived from the verbal communication, thus opening up the possibility of humans interacting or communicating intuitively with computers or robots by indicating objects or pointed locations either in the 3D space or on the screen. However, it is a challenging task to estimate the 3D hand pointing direction automatically and reliably from the streams of video data due to the great variety and adaptability of hand movement and the undistinguishable hand features of the joint parts. Some previous work show the success in hand detection and tracking using multi-colored gloves [16] and depth-aware cameras [8], or background subtraction [14], color-based detection [7, 8], stereo vision based [2, 4, 18] or binary pattern based [5, 10] hand feature detection. However, the big challenge remains for accurate hand detection and tracking in terms of various hand rotations.

Motivated by the recent advances of feature detection [1, 5, 6, 8, 10, 11, 12, 19], we propose to develop a novel technique to estimate pointing direction based on two orthogonal-view cameras. Here, we only focused on the gesture of hand pointing. We setup two regular cameras in orthogonal positions, one on the top of the user, and the other to the left side. Figure 1 shows the major components of the system. Unlike binary-pattern based approaches [5, 10] which are limited to a certain degree of hand rotation, we propose a hand-image warping approach to transform the original hand image to a polar-coordinate plane in order to make the hand detection invariant to orientation. We apply two cascade detectors based on binary pattern features and AdaBoost [15] for two views' hand region detection. Then we used the Active Appearance Model (AAM) [3] to track the finger points to identify the direction of hand pointing. We extend the idea of AAM face tracking to track landmark features of hands. We are able to infer the 3D orientation of a pointing finger in 2D space. This is done via two simultaneous captures of the hand. There is correspondence between the points in the top and side views. Using this correspondence between the points allows us to infer the (x, y, z) coordinates of those points. Once we have the 3D coordinates we use two points along the finger to draw a vector in 3D space, resulting in the orientation of the finger. Following sections will describe each component individually.

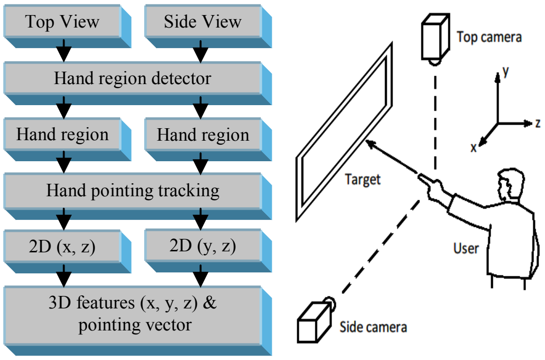

**Figure 1: System composition for pointing estimation**

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.916

## 2. Two-Views' Hand Region Detection

Hand region detection is the first step towards the pointing direction estimation. Figure 2 shows the diagram of the hand region detection for both views.

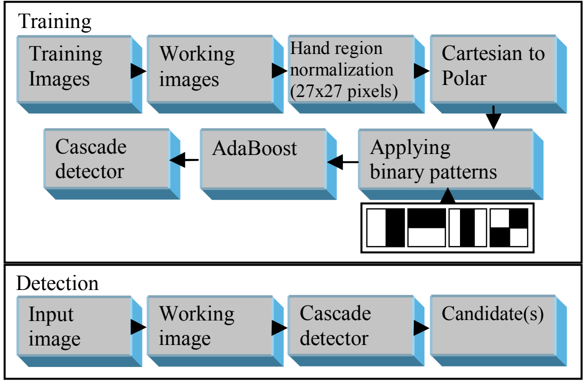

**Figure 2: Components of hand region detection**

Motivated by the success of face detection developed by Viola-Jones [15] using Haar-like features and an AdaBoost cascade detector, we extend the features to hand regions detection for the pointing gesture. The binary patterns used in [15] describe the features within facial regions with no background involved. However, the most significant features of a hand pointing gesture are the shape of the hand rather than internal hand shades and textures. As we need features to describe the shape of the hand pointing, we will have to involve the background. Unfortunately, various backgrounds may cause instability for hand detection. Here we propose a simple approach using color channel arithmetic to reduce the influence of backgrounds. By observing that the skin color has much stronger red component than the green and blue components, we compute the image by

$$I(x,y) = R(x,y) - Max\{G(x,y), B(x,y)\} \qquad (1)$$

This simple process can roughly highlight the skin area. As a result, a working image $I$ is generated, based on which subsequent operations using the binary pattern approach can be carried out effectively. Figure 3 illustrates an example of the generated two views' working images.

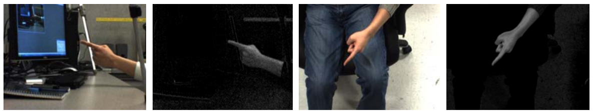

**Figure 3: Working images (2nd and 4th images from left)**

### 2.1. Hand image warping using polar coordinates

Some existing work has applied pre-designed binary patterns for hand detection successfully [5, 10]. However, the detection is still sensitive to the variation of hand orientations. The report in [5] shows that only $15^o$ of hand rotation can be detected by applying the Viola-Jones-like approach [15]. To improve the orientation invariance to hand region detection, we propose to warp the hand image from Cartesian coordinates to polar-radial coordinates. To do so, we use the center of the window as a pole ("o"), and the polar angle (θ) at 0 degree is determined by the position of the wrist (Figure 5). The radius (r) of the polar axis is determined by the window size.

Since a hand is always connected with its corresponding wrist, in order to estimate the wrist position, we divide the 27×27 hand image into 3×3 blocks as shown in Figure 4.

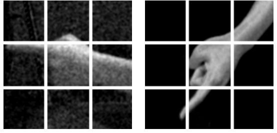

**Figure 4: Hand wrist estimation by 3×3 blocks division**

The fist is contained in the central block, and the wrist is located at one of the surrounding 8 blocks. Due to the strong correlation of skin colors between hand and wrist, the average color of the block containing the wrist is the most similar to the average color of the central block among the 8 surrounding blocks. Then we are able to identify the position of the wrist by comparing the average color of the 8 blocks and the central block.

After the position of the wrist is determined, we use this position as the 0 degree polar coordinate, and convert the image from Cartesian (x, y) to Polar coordinates (θ, r). Figure 5 shows examples of the image warping from both views. As we can see, the converted images have similar appearances regardless of hand orientations rotated in the image plane.

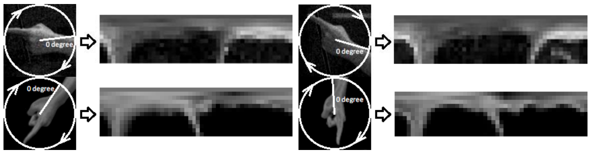

**Figure 5: Image conversion from Cartesian to Polar coordinates. Upper row: side view; Lower row: top view. The resolution of warped images is 48×12.**

### 2.2. Binary pattern based hand detection

After the image conversion, we apply the binary patterns as shown in Figure 2 (four black-white patterns) to the warped image in (θ, r) coordinates. Figure 6 illustrates an example of the binary patterns overlapping on the warped image.

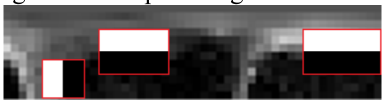

**Figure 6: Example of applied binary patterns**

Similar to the procedure used in [15], our hand detector performs following three operations: (1) integral image generation, (2) Haar-like features generation using the above binary patterns, and (3) building cascade detector using AdaBoost.

After the detector has been built it scans the input image in a brute-force way. All sub-windows with different size and position in the image will be input to the detector. Once a sub-window has passed the detector, it will be marked as a candidate.

## 3. Two-Views' Hand Feature Tracking

Given the detected hand regions, we are able to track hand features in the limited search regions. We apply an Active appearance model (AAM) [3] to track 14 pre-defined feature hand-points on both top view and side view. AAM is a method of matching statistical models to images developed by Cootes *et al.* [3]. A set of landmark images are used to create the training set. The model parameters that control the shape and gray-level variation are subsequently learned from this training set.

The landmarks selected for the training set represent the shape of the object to be modeled. These landmarks are represented as a vector and principal component analysis is applied to them. This can be approximated with the following formulas: $x = \bar{x} + P_s b_s$ for shape and $g = \bar{g} + P_g b_g$ for texture. In the shape formula $\bar{x}$ is the mean shape. $P_s$ represents the modes of variation and $b_s$ defines the shape parameters. In the texture formula $\bar{g}$ is the mean gray level. $P_g$ represents the modes of variation and $b_g$ defines the grey-level parameters.

We use AAMs to create a statistical model of the hand from two orthogonal views via a simultaneous capture. We create a separate appearance model for each view, and track the hand in two views separately. To create the hand shape and gray-level models we chose 14 landmarks for the training set images. These landmarks for the top and side views can be seen in Figure 7. Note that the hand detection of the previous stage allows us to narrow down the search region for fitting our model to the hand, thus reducing the time for finding a correct fit.

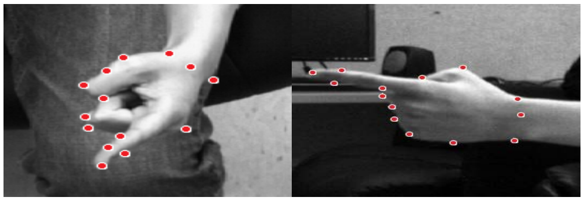

**Figure 7: 14 landmarks for top view and side view.**

## 4. Estimation of 3D Pointing Direction

Since the two views of hands are tracked separately with different models, we are able to create the best fit for the corresponding hand in each frame. There is correspondence between multiple landmarks in the separate views. Those landmarks, most notably on the finger, allow us to infer the 3D coordinates from 2D coordinates, and infer the 3D orientation of the finger. For one point that has correspondence between the two models, we can use the top view as the (x, z) coordinate and the side view as the (z, y) coordinate. We can then combine both of the views to infer the (x, y, z) coordinate for that tracked landmark. Since the z coordinate may not be the same in both of the views, we take the average of both values to give us a new z coordinate.

Once we have the 3D coordinates of the tracked points we take two points on the finger that are "connected by a line" to create a vector that points in the direction of the finger. The two points selected are near the top and bottom of the pointing finger. These were selected as they appear to give us the most reliable vector in determining the orientation of the finger. The other landmarks are used to create a better fit for the AAM search, as well as for future modeling of hand details. This vector is shown in Figure 9 via the line pointing from the finger.

## 5. Experiments and Evaluations

Our system is setup with two regular cameras, one being a top view and the other a side view. The system works with a resolution of 640x480. To train two detectors for two views separately, we selected 107 positive image samples and 160 negative samples for the top view of hands, and 128 positive samples and 145 negative samples for the side view. Figure 8 shows examples of training images.

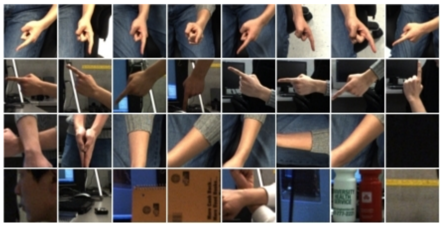

**Figure 8: Examples of training images. Row 1 and 2: positive samples of top view and side view, respectively. Row 3 and 4: negative samples for both views.**

In the training stage, we applied the binary patterns to the converted image in the polar coordinate system, and generated over 5,000 features for each sample. Then, two views cascade detectors are built based on the feature selection by AdaBoost. In the testing stage, after the input image is converted to the working image, each detector scans the working image in each view separately. Note that during the hand region search, the integral image is computed locally on the warped image in the polar coordinate system. The experiments are conducted in our lab environment. The hand motion is in the range of [-60°, +60°] for both pan and tilt. Figure 9 shows an example of the estimated pointing vectors (red lines) along with the detected hand regions (green blocks) and the tracked 14 feature points (black dots). Table 1 shows the degrees of rotation of the finger pointing in Figure 9 and their corresponding normal vectors $\vec{n}$ .

Comparing the detected hand regions with the manual selected ones, we achieved 90% and 91% correct detection rate for the top-view (691 images) and side view (829 images), respectively. In addition, the estimated hand pointing orientations are also compared to the physically measured hand orientation during the time of capture. Among 7,600 frames, 6,916 frames show less than 5-degree difference between two data sets. The correct pointing rate is 91%.

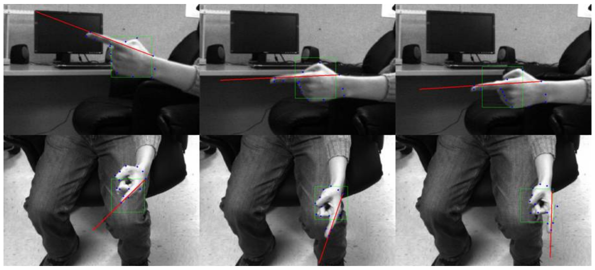

**Figure 9: Sampled frames of a testing video showing detected hand regions (green blocks), tracked points (blue dots), and the estimated pointing vectors (red lines) from a side view (upper row) and a top view (lower row).**

**Table 1: Degrees of rotation for vectors in Figure 9**

| | | | |
|---|---|---|---|
| **Tilt (upper row)** | 20.6° | -2.57° | -3.81° |
| **Pan (lower row)** | -45.81° | -18.94° | -1.42° |
| $\vec{n}=(\vec{x},\vec{y},\vec{z})$ | (-0.58, 0.20, -0.56) | (-0.23, -0.03, -0.67) | (-0.02, -0.06, -0.99) |

## 6. Conclusions

In this paper, we presented a newly developed system for hand pointing direction estimation with a certain degree of robustness to hand rotation. By using two camera views to capture a hand we are able to track and model the hand via active appearance models. Using the tracked points from the AAM allows us to infer the 3D orientation of the pointing finger due to the correspondence between points on the finger of both views. Our future work is to improve our system by addressing issues with respect to illumination variations and global integral image computation in order to increase the robustness and speed of the detection process. In addition, we will develop a model for pointing cursor determination on a screen, and evaluate the pointing direction and its error by measuring the difference between the projected positions and the expected positions on the screen. Moreover, 3D hand models will be captured using our 3D imaging system for further evaluation of 3D pointing accuracy.

## 7. Acknowledgement

This material is based upon the work supported by the Air Force Research Lab at Rome, NY.

## References

[1] L. Clinque, et al, Fast viewpoint-invariant articulated hand detection combining curve and graph matching, *IEEE FGR* 2008.

[2] C. Colombo et al., Visual capture and understanding of hand pointing actions in a 3-D environment, *IEEE Trans. on SMC-B* 33(4), 2003. pp. 677-686.

[3] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE PAMI*, 23(6): 681-685, 2001.

[4] N. Jojic et al, Detection and estimation of pointing gestures in real-time stereo sequences, In *IEEE FGR'00*.

[5] M. Kölsch and M. Turk. Analysis of rotational robustness of hand detection with a viola-jones detector. *ICPR* 2004.

[6] I. Fratric and S. Ribaric, Real-time model based hand localization for unsupervised palmar image acquisition, Proc. ICB 2009.

[7] M. Lee, D. Weinshall, et al. A computer vision system for on-screen item selection, *IEEE CVPR* 2001.

[8] C. Manders, F. Farbiz, et al. Robust hand tracking using a skin tone and depth joint probability model. *FGR,* 2008.

[9] T. Moeslund and E. Granum, A survey of computer vision-based human motion capture, *CVIU* 81(3), 2001.

[10] T. Nguyen, N. Binh, and H. Bischof. An active boosting-based learning framework for real-time hand detection. *IEEE FGR,* 2008.

[11] K. Oka, Y. Sato, and H. Koike, Real-time fingertip tracking and gesture recognition, *IEEE Computer Graphics and Applications*, 2002.

[12] C. Park, M. Roh, and S. Lee, real-time 3D pointing gesture recognition in mobile space, *IEEE FGR* 2008.

[13] V. Pavlovic and T. Huang, et al, Visual interpretation of hand gestures for HCI: review, *IEEE Trans. PAMI,* 1997.

[14] A. Utsumiy, et al, Hand detection and tracking using pixel value distribution model for multiple-camera-based gesture interactions. *IEEE CVPR*, 2001.

[15] P. Viola and M. Jones. Robust real-time face detection, *International J. of Computer Vision* 57(2)137–154, 2004.

[16] R. Wang and J. Popovic, Real-time hand-tracking with a color glove, *SIGGRAPH 2009.*

[17] Y. Wu and T. Huang, Vision-based gesture recognition: A review, In *IEEE 3rd Gesture Workshop*, 1999.

[18] Y. Yamamoto et al, Arm-pointing gesture interface using surrounded stereo cameras system, In *ICPR 2004*.

[19] B. Stenger, A Thayananthan, P. Torr, and R. Cipolla, Model-based hand tracking using a hierarchical Bayesian filter, IEEE Trans. PAMI, 28(9): 1372-1384, 2006.