

LETTER

A Real-Time Hand Pose Recognition Method with Hidden Finger Prediction

Min-Young NA[†], Nonmember and Tae-Young KIM^{†a)}, Member

SUMMARY In this paper, we present a real-time hand pose recognition method to provide an intuitive user interface through hand poses or gestures without a keyboard and a mouse. For this, the areas of right and left hands are segmented from the depth camera image, and noise compensation is performed. Then, the rotation angle and the centroid point of each hand area are calculated. Subsequently, joint points and end points of a finger are detected by expanding a circle at regular intervals from a centroid point of the hand. Lastly, the hand pose is recognized by matching between the current hand information and the hand model of previous frame and the hand model is updated for the next frame. This method enables users to predict the hidden fingers through the hand model information of the previous frame using temporal coherence in consecutive frames. As a result of the experiment on various hand poses with the hidden fingers using both hands, the accuracy showed over 95% and the performance indicated over 32 fps. The proposed method can be used as a contactless input interface in presentation, advertisement, education, and game applications.

key words: hand pose recognition, hand-based user interface, hand model

1. Introduction

Ubiquitous environment systems are being widely developed thanks to the advancement of information technology. In the systems, it is necessary to provide a user-friendly input interface without a keyboard and a mouse that does not constrain the user's action but allows information to be transmitted.

The study on an intuitive user-friendly input interface such as multi-touch interface, gesture recognition interface using sensor, and vision-based contactless interface has been actively performed in recent years. However, when the user uses a multi-touch interface or a sensor, it requires separate equipment. Thus, it is difficult to apply to various applications, which of course functions as a disadvantage. On the other hand, vision-based input interface can be used as a contactless type out of hardware in various applications including education, games, presentations, etc. by recognizing user's hand poses or gestures. For this, the studies have been carried out in various ways [1]–[10].

In this paper, we propose a method of enabling the recognition of a variety of hand poses in real time. The method is able to find out the location of the hidden fingers even if the hand moves through gestures, rotation, tilting, etc. after the fingers are hidden. In addition, the algorithm is relatively simple, so it has strength in that the speed is quick

and the accuracy is also high. This method can be used as a contactless input interface.

2. Algorithm of Recognizing Real-Time Hand Poses, Predictable Hidden Fingers

When the user takes a hand pose, temporal coherence exists between two consecutive frames with a short renewal cycle. In other words, the segmented hand images between the two frames aren't significantly different, so it is used to recognize the current hand pose, based on the hand model obtained previously. Figure 1 shows the whole process of our algorithm.

2.1 Hand Segmentation and Noise Compensation

When the user takes action of waving the hand from side to side toward the Kinect camera by spreading the five fingers of both hands, each hand tracking point is detected. Based on the point, we segment each of the hands in the camera image by carrying out 8-way navigation based on this point and obtaining the average of the depth values of pixels until the boundary points of the hand and the background, whose depth values rapidly change, come out. The segmented hand image causes jittering to occur to its outlying portion due to

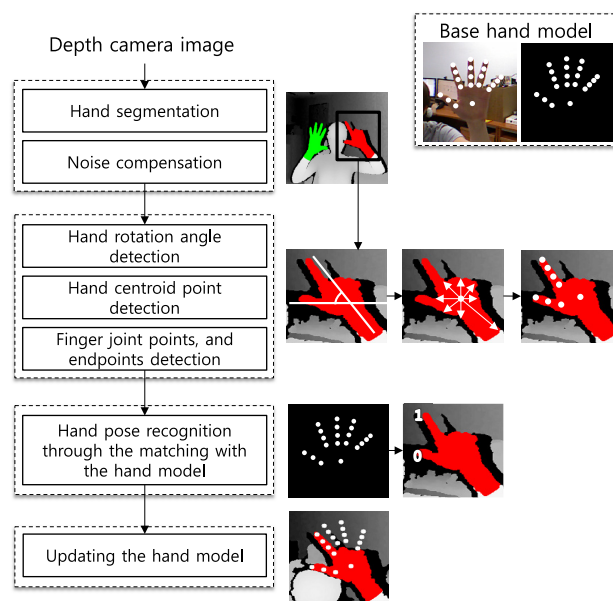


Fig. 1 The proposed algorithm.

Manuscript received December 18, 2012.

Manuscript revised April 23, 2013.

[†]The authors are with SeoKyeong University, Seoul, 136–704 Korea.

a) E-mail: tykim@skuniv.ac.kr (Corresponding author)

DOI: 10.1587/transinf.E96.D.2170

the characteristics of the camera based on the depth value. This result lowers pose recognition accuracy. In our method, we remove the noise of the outskirts of the hand for the noise compensation of the hand image by performing the outline approximation applying the DP (Douglass-Peucker) algorithm, as shown in Fig. 2

2.2 Detection of Hand Rotation Angle and Hand Centroid Point

In the following step, we detect the hand rotation angle and the hand centroid point. First, we conduct line fitting by performing the RANSAC (Random Sample Consensus) algorithm [12] starting the random two points in the hand image. After that, we obtain the rotation angle of the hand by applying the Eq. (1) for two random points (p_1, p_2) on the line approximated.

$$\begin{aligned} \text{radius} &= \text{atan2}((p_1.y - p_2.y), (p_1.x - p_2.x)) \\ \text{rotation angle} &= (\text{radius} \times 180)/\pi \end{aligned} \quad (1)$$

Figure 3 shows the results of conducting the line approximation in a variety of hand pose images. Figure 4

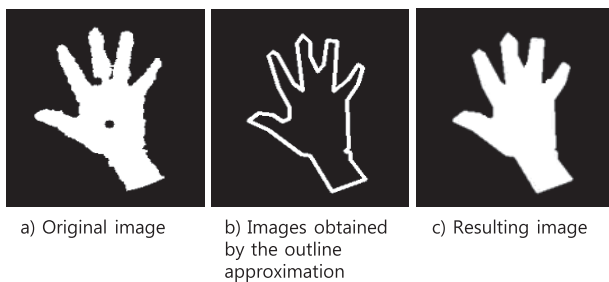


Fig. 2 Noise compensation.

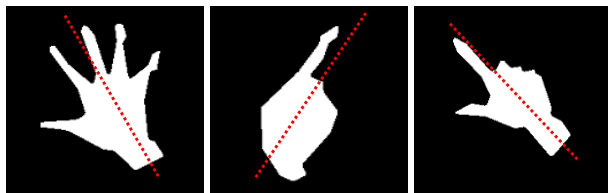


Fig. 3 Line fitting in a variety of hand pose images.

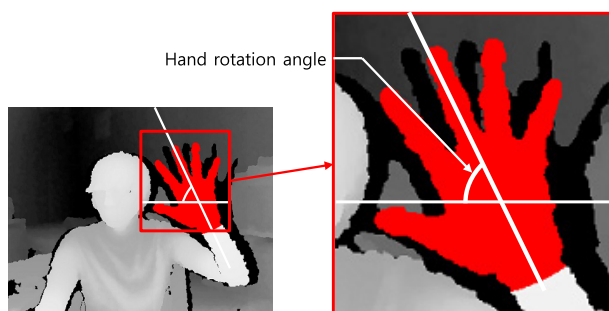


Fig. 4 Calculation of the hand rotation angle based on the line - fitted straight line.

shows the process of calculating the hand rotation angle.

In the next step, we obtain the hand centroid point using a distance map. As shown in Fig. 5, we obtain the distance map to save the shortest distance value into the corresponding pixel location by performing eight-way navigation for each pixel in the hand area. Then, the pixel with the largest one of values stored in the distance map becomes the hand centroid point.

2.3 Detection of the Finger Joint Points and Endpoints

Figure 6 shows the process of detecting the finger joint points and endpoints for hand pose recognition. First, we expand the circle at regular intervals on a basis of the detected hand centroid point and try to find out intersections of the white and black pixels of the hand by drawing the circle. Then, we determine the intersected points as a boundary candidate of the finger and connect them to obtain the midway points. In this process, the midway points may occur to the wrist and palm. Thus, it is needed to conduct removing other points other than the fingers. For this, we remove points on the wrist by calculating the angle between each point and the centroid point of the hand based on the rotation angle of hand. Then, we perform clustering the points with each other, where the angle between each point and the hand

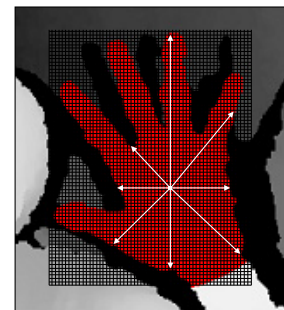


Fig. 5 Detection of the hand centroid point based on the distance map.

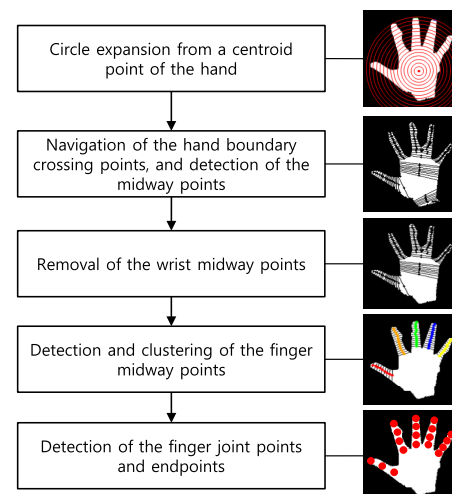


Fig. 6 Process of detecting the finger joint points and endpoints.

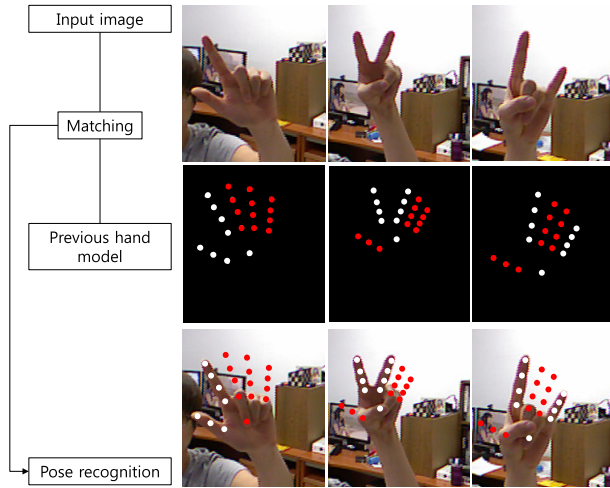


Fig. 7 Hand pose recognition through matching with the hand model.

centroid point is located within the critical angle of each finger. Then, we determine the relatively short part is not a finger so remove it. After then, we obtain the joint points and endpoint for each finger based on the divided points.

2.4 Hand Pose Recognition by Matching with the Hand Model

In the first frame, the user opens the five fingers all and comes to generate the initial hand model based on the information of the hand calculated in Sect. 2.3. After that, from the second frame we perform matching between the current hand information and the hand model made in the previous frame by comparing the differences in the distances and angles between joint points and end point of each finger and the centroid point of the hand. Then, we recognize the hand pose and update the previous hand model into the hand model for the current frame. Thus, the hand model indicates the finger information for each finger stretched most recently.

The size or rotation angle of the segmented hand image may change flexibly in each frame. Thus, before matching, we rotate the hand model to ensure corresponding to the rotation angle of the hand in the current image. Then, we normalize and scale it according to the hand size in the image. After that, we conduct the matching job.

Figure 7 shows the process of obtaining the pose of the hand through matching the hand model. When the finger is hidden and then appeared as shown in the Fig. 7, it is very difficult to accurately identify the finger if it is neither a thumb nor a little finger. However, our method will be able to recognize which finger it is, so it would be possible to recognize a variety of hand poses.

3. Results

This method was developed in the environment of the Intel Core I5 2500 Processor and 4 GB RAM, VS 2005. For this,

Table 1 User input interface definition according to hand poses.

Pose	Gesture	Pose	Gesture
	Left Button Click		Scroll up
	Zoom		Scroll down
	Rotate		Close



Fig. 8 Manipulating a viewer via a hand-based input interface.

we used OpenCV 2.1 and OpenNI. For a camera, we used MS Kinect. The image size of the used camera is 640×480 . As the distance between the Kinect and the user increases more, the resolution of the detected hand gets smaller and the Jittering phenomenon becomes serious. Therefore, the recognition rate will fall sharply. In this paper, experiments were carried out at the point of 60-90 cm away from the Kinect.

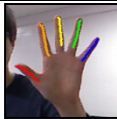
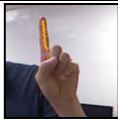
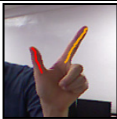
To test the proposed method, we defined a hand-based input interface to control application programs through hand gestures defined a series of hand poses, as shown in Table 1.

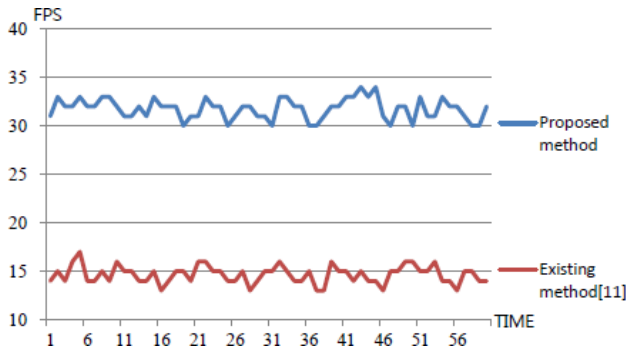
Figure 8 shows the results of applying the previously-defined hand based input interface by implementing a multimedia viewer for smart devices. The user could freely use such functions as mouse left and right clicking, zooming, image flipping, etc. using both hands. In particular, the method proposed in this paper enables to identify the finger of each hand. Thus, it was observed that more various hand poses are used as a user input interface.

To measure the hand pose recognition used in our method, three testers used the user interface of the application program 50 times and repeated this process 2 times. The results showed 95% of recognition rate on average.

We conducted a comparative experiment with the existing method [11]. In the method, the information of 4 major characteristics (hu moment, angle count, skin color angle, non-skin color angle) of each hand pose is saved as a pre-processing step. After that, the hand pose is recognized by template matching between current input image and pre-

Table 2 Recognition rates.

Pose						
Proposed method	100%	93%	95%	94%	92%	98%
Existing method[11]	96%	90%	93%	90%	86%	96%

**Fig. 9** Speed comparison with the existing method [11].

stored images. Each of three experimenters performed the pose defined in Table 2 100 times.

As a result of the existing method [11], a matching error occurred between similar poses. On the contrary, our method showed higher recognition accuracy because the hand model is newly updated through the feature information obtained in each frame and is used for matching in the next frame. For the hand pose recognition, as the number of stretched fingers was fewer, the recognition rate was lower with the lack of features used for matching. In addition, the results showed a high recognition rate for the front and the rotation. But it was observed in this method that when the outline overlaps or is screened, the recognition rate is relatively low.

In terms of speed, as shown in Fig. 9, the existing method [11] showed that about 15 minutes of pre-processing time was needed and the performance reached 15 fps on average. Contrarily, our method indicated 32 fps on average due to the less number of feature points required for pose recognition and no need for prior learning.

4. Conclusion

We present a real-time hand pose recognition method to provide hand poses or intuitive user input interfaces. This method enables the user to predict the hidden finger of the current frame using the hand model through the temporal coherence between consecutive frames. Based on this, the user therefore can perform the robust real-time hand pose recognition even in a situation of rotation and tilting. It is expected that a contactless user interface to control a smart TV

will be presented by adding a gesture recognition scheme through machine learning to a hand pose recognition method according to the results of this paper in the near future.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0065960)

References

- [1] V.I. Pavlovic, R. Sharma, and T.S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.19, no.7, pp.677–695, 1997.
- [2] D. Geer, "Will gesture recognition technology point the way," *Computer*, vol.37, no.10, pp.20–23, 2004.
- [3] M. Tosas and B. Li, "Virtual touch screen for mixed reality," *Proc. European Conference on Computer Vision, Lecture Notes in Computer Science*, Prague, Czech Republic, vol.3058, pp.48–59, May 2004.
- [4] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla, "Hand pose estimation using hierarchical detection," *Proc. European Conference on Computer Vision, Lect. Notes Comput. Sci.*, Prague, Czech Republic, vol.3058, pp.105–116, May 2004.
- [5] M. Tang, "Hand gesture recognition using microsoft kinect," *CS228*, 2010.
- [6] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst. Man Cybern. C: Appl. Rev.*, vol.37, no.3, pp.311–324, 2007.
- [7] M. Ko and J. Yoo, "Histogram based hand recognition system for augmented reality," *KIICE*, vol.15, no.7, pp.1564–1572, 2011.
- [8] M.-Y. Na, H.-J. Kim, and T.-Y. Kim, "An illumination and background-robust hand image segmentation method based on the dynamic threshold values," *KMMS*, vol.14, no.5, pp.607–613, 2011.
- [9] S. Malassiotis and M.G. Strintzis, "Real-time hand posture recognition using range data," *Image Vision Comput.*, vol.26, no.7, pp.1027–1037, 2008.
- [10] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.2, pp.126–133, June 2000.
- [11] Y. Liu, L. Zhang, and S. Zhang, "A hand gesture recognition method based on multi-feature and template matching," *Procedia Engineering*, vol.29, pp.1678–1684, 2012.
- [12] M.A. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol.24, no.6, pp.381–395, 1981.