Hand Tracking with a Near-Range Depth Camera for Virtual Object Manipulation in an Wearable Augmented Reality

Gabyong Park, Taejin Ha, and Woontack Woo

KAIST UVR Lab., S. Korea
{gypark,taejinha,wwoo}@kaist.ac.kr

Abstract. This paper proposes methods for tracking a bare hand with a near-range depth camera attached to a video see-through Head-mounted Display (HMD) for virtual object manipulation in an Augmented Reality (AR) environment. The particular focus herein is upon using hand gestures that are frequently used in daily life. First, we use a near-range depth camera attached to HMD to segment the hand object easily, considering both skin color and depth information within arms' reaches. Then, fingertip and base positions are extracted through primitive models of the finger and palm. According to these positions, the rotation parameters of finger joints are estimated through an inverse-kinematics algorithm. Finally, the user's hands are localized from physical space by camera-tracking and then used for 3D virtual object manipulation. Our method is applicable to various AR interaction scenarios such as digital information access/control, creative CG modeling, virtual-hand-guiding, or game UIs.

Keywords: Hand Tracking, HMD, Augmented Reality.

1 Introduction

Today, as cameras and HMDs are smaller and lighter, wearable computing technology is garnering significant attention. There are many interface systems for obtaining digital information about the object, space, or situation in which a user is interested. Of the various user interfaces, the hand is naturally anticipated as a major focus for wearable computing technology.

In wearable computing, considerable research has advanced the development of natural user interfaces with a hand. [1–5] proposed a hand-tracking and pose estimation technology, through pattern recognition or tracking a hand after initializing, which is appropriate in a fixed camera environment. On the other hand, [7-8] proposed a system that is suitable for moving-camera-view situations. The algorithm for bare hand-based user interface in fixed environment of a camera is not appropriate to HMD wearable interfaces like our system. The algorithm has difficulties with background learning, because of the camera's movement; thus, it cannot estimate hand pose well. In moving-camera-view scenarios with HMD, as far as we are aware, researchers have used mainly color information. But this has limits in 3D motion recognition.

R. Shumaker and S. Lackey (Eds.): VAMR 2014, Part I, LNCS 8525, pp. 396–405, 2014.

For the interactions with virtual objects registered in real world, using a hand, the system has to coordinate successfully between the real object and the hand.

We propose methods for a user wearing HMD to manipulate the virtual object with their own hand in a wearable AR environment, without a desktop-based interface with a mouse and keyboard. Concretely, from skin color and depth information, it extracts positions of the tips and bases of fingers based on its models of the finger and palm. Then, to estimate the rotation parameters of a finger's articulations, an inverse kinematics algorithm is used, which considers the damping ratio using the position of a fingertip and the base of the finger. Finally, for AR applications, gesture-based interaction is processed with coordinates between hand and real object.

We focus on the accuracy and stability of gesture estimation. For this, we model a finger and palm through a convex body. The features of a palm and a finger are extracted stably. This enables the detection module to be robust in nearly real time. In addition, through the hand gestures, virtual objects are manipulated. Figure 1 shows our application scenario. A user who wants to go on a trop can select virtual objects (e.g., area) augmented on the virtual globe object. The user can adjust the virtual object's position, orientation, or size for detailed observation.

This paper is composed of the following components: Section 2 introduces related works about hand tracking, and Section 3 describes our systems with hand-tracking. Next, Section 4 shows the experiment for accuracy and stability of our hand-tracking module. Finally, we introduce the conclusions regarding our system and options for future work in this area.

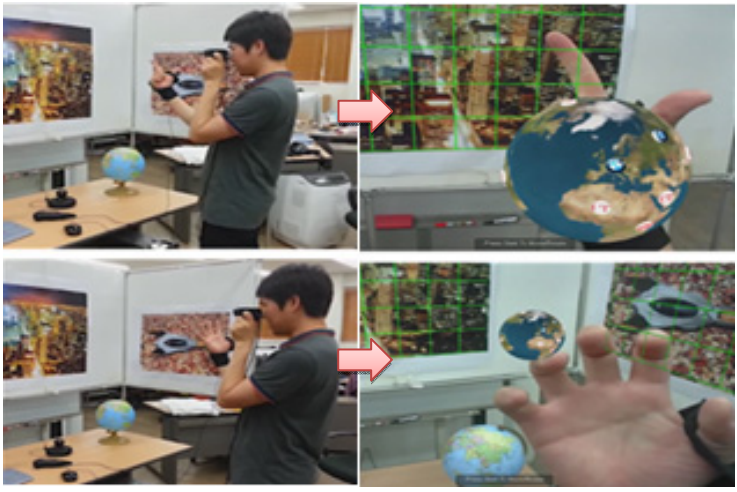

Fig. 1. Bare hand user interaction on video see-through HMD with a RGB-D camera

2 Related Works

There is extensive extant research on hand tracking. To track the motion of a hand, various methods have been developed. A hand has 26 Degrees of Freedom (DOF) [10].

Using this fact, model-based approaches were developed by [1-2]. This proposed 3D tracking of hand articulations by using RGB-D camera like Kinect. This hypothesizes the hand motion with Particle Swarm Optimization (PSO), and estimates tracking accuracy through pixel estimations of the color and depth map. [2] has expanded upon this version, applying the algorithm to two hands. This shows robust tracking results for each hand and its fingers. [1-2] performed detection procedures through initializing. By a model-based approach, this shows very robust tracking of the fingers' self-occlusion.

A feature-based approach was developed by [3-4]. This proposed a method for estimating the gestures of the hand by recognizing patterns from off-line learning. This system sets up the RGB-D camera in a position in which it is looking down, and requires a user to wear a special glove with a particular, identifiable pattern [3]. By this method of pattern recognition, up to two hands are tracked. The user can interact with virtual objects registered to the real world. In the present, furthered version, a method is proposed that does not require special gloves [4]. These methods take advantage of shape features; but this method, in certain situations, has difficulty recognizing certain motions.

Another approach, based on artificial neural networks, is proposed by [5]. This develops the system to track the torso and hands with a Self-Organizing Map (SOM). This does not require off-line learning, and it enables the algorithm to operate in an ARM-based platform. Thus, this offers a simple and speedy algorithm to track hand motion.

Using detection of fingertips and a palm, methods were developed by [6–9]. [6] proposed a finger-tracking method suitable for a wearable device, which is equipped on a wrist. Through depth information based on Time of Flight (TOF), it has excellent finger-tracking accuracy. Because the device is equipped on the wrist, self-occlusion of fingers does not incur. [7] recognized a hand's pose with polar coordinates. The number of fingers is counted by these polar coordinates. This method is suitable for wearable environments because it operates well only if the hand object is segmented well. [8] proposed the algorithm for acquiring coordinates of a hand from a camera, using points of the fingers and the palm. This does not need any marker for augmented-reality interaction and is, thus, useful for wearable computing applications. [9] is the near-range RGB-D camera, the same device that we use. In the hand-tracking module of that device, the fingertip and center of a palm are tracked. This operates in real time and robustly.

As we described in the related works about hand-tracking system, some researchers have focused on full-motion tracking of a hand. For this approach, they gave some constraints to the system, like a fixed camera and special glove [1–6]. Unless they have exceptional accuracy, for wearable AR environments, the camera's view should coincide with the view of a user. This configuration generates both a changing environment as well as difficulties of initializing for detection.

Although not tracking full DOF motion, some researched methods [7–9] are suitable for wearable AR applications, using the hand. [7-8] do not take enough advantage, however, of a hand's depth information, and so they cannot track a hand's 3D motion very well. In the module included in [9], especially when tracking bending fingers, the tracking accuracy is also problematic, with limited hand pose.

We focus on the hand-motion-tracking for wearable AR environments. Tracking is conducted robustly, using the featured information of fingertip, base, and palm, which are extracted easily in spite of the moving-camera view. In other words, through RGB and depth information, regardless of the changing environmental scene, a hand's features are extracted stably. Specifically, this makes it possible to estimate gestures used frequently in our daily life, such as pointing, pinching, grasping, translation, and rotation. Finally, for interaction with virtual objects adjusted to the real world, relative coordinates between a hand and the real object are calculated. On this basis, a user can manipulate virtual objects in a wearable AR environment.

Our contributions are as follows:

- Hand-tracking is conducted robustly regardless of the changing scene, considering the range a user can reach with an arm. It is also a suitable configuration for wearable augmented reality interaction, because the camera view coincides with that of the user.
- The fingertips, base of a finger, and a palm are detected robustly, regardless of the changing environment. This feature is used to track a hand and estimate gesture of it, used frequently in our life.

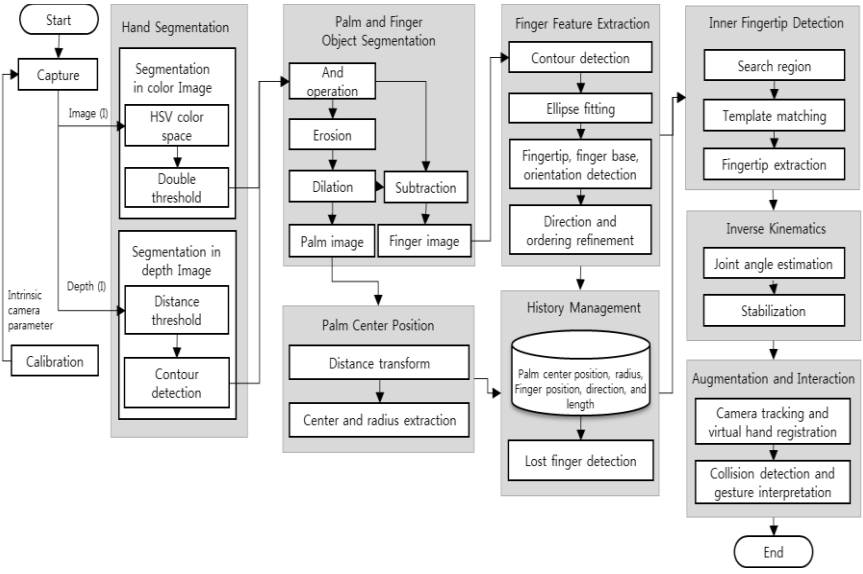

Fig. 2. Proposed hand tracking algorithm for manipulating virtual objects in wearable AR

3 System

Figure 2 shows a block diagram of our system. This is the algorithm that tracks a hand for 3D user hand interaction with a RGB-D camera in wearable AR environments. First, from RGB image and depth image through the camera, hand–object segmentation is conducted according to HSV color space and fingers' objects; the palm's is

segmented through morphological operation. When modeling the fingers and a palm, featured information includes the center/radius/direction of a palm, the position of a fingers' tip and base, and ordering. In addition to that, using a depth template of fingertips and motion information of the fingers and a palm, in the case of bending fingers inward, 3D position of fingertips can be extracted. After that, from the extracted 3D position of fingertips and their bases, rotation parameters are estimated using an Inverse Kinematics (IK) algorithm. Finally, virtual objects are registered to a real object, and coordinates between a virtual object and the real hand are derived. In this system, a user can interact with virtual objects with their own hand.

3.1 Palm and Finger Object Segmentation

From the RGB and depth image, the objects composing a hand are segmented [11]. First, for robust segmentation according to light condition, the RGB color space is converted to HSV color space. The skin color space is attained through a double threshold about S and V components. Additionally, depth segmentation by the region that a user can reach with the hand makes it possible to segment hands and objects robustly and easily. To this end, depth values in the depth image showing the distance from the camera's image plane to a fingertip are stored in the map. We configure this depth threshold value as 60 cm. Finally, from the skin color and depth image maps, each segmented an image is overlapped.

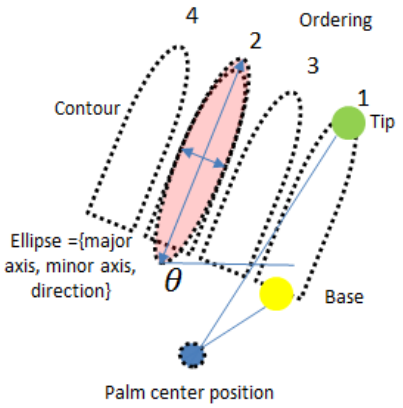

Fig. 3. Feature extraction of fingers and a palm

From this segmented hand image, the image is further segmented into one for the fingers and one for the palm to estimate a hand's subspecialized poses. For this step, a hand image is segmented through simple morphological operations such as erosion, expansion, and subtraction [9].

A palm's center position and radius are calculated by a distance transform [13]. Later, for estimating the poses of fingers by the IK algorithm, the positions of the fingers' tips and bases are needed. Finger objects are modeled by ellipse-fitting. As shown in Figure 3, the finger's base point of is calculated by minimizing of the distance from a point in a palm's circle model to a point in a finger's ellipse model.

This method is advantageous, allowing as it does the stable extraction of the fingers' base, even when the fingers are bending.

Through the above procedure, when fingers are bent inward to the palm, the position of the fingertip is not detected, although one of the bases is found. This is because the points of ellipse models do not include the fingertip in the image plane. Given this problem, we cannot recognize many gestures on the basis of fingers' motions. Thus, we should extract the fingertips' positions by another approach, such as the use of depth information.

To detect the position of the fingertips, we take care of their motion information. When one bends a finger inward toward the palm in the present frame, the convex body modeled is not detected. So, through the trace of a fingertip stored in some previous frames is known the fact that this fingertip is in the region of the palm. The image's region is set according to the direction in which the fingertip moved, and the Zero-mean Normalized Cross Correlation (ZNCC) [14] is calculated, using the depth template stored in the off-line procedure. As figure 4 shows, one point having a high score is extracted, which estimates the position of the fingertip detected. This point and the base point detected from model-fitting are used in experiment as input parameters for an inverse-kinematics algorithm.

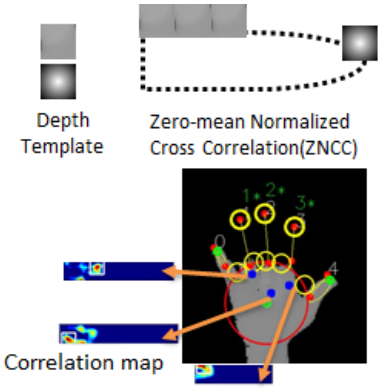

Fig. 4. Fingertip detection using depth template

3.2 3D Hand Model Reconstruction Using Damped Least Squares Based Inverse Kinematics

To estimate the rotation poses of fingers, we use an IK algorithm. We apply it to our system, which is a damped least-square-based inverse-kinematics algorithm that [12] has proposed. This method, unlike other algorithms, can estimate rotation parameters stably by controlling the damping ratio. To take advantage of this algorithm, the position of end-effector and reference like fingertips and finger's base is detected. From [section 3.1] we proposed, the positions are considered as input, and rotation parameters are estimated. Like figure 5, we model virtual hand controlled from camera image for estimation.

We denote $\vec{s}$ as position vector from base point and $\vec{t}$ as target position vector detected from camera image, θ as a joint's rotation parameter, λ as damping ratio

parameter, and set L1, L2, L3 as the joint's length. Thus, the IK algorithm solves optimizing problems as follows:

$$\Delta\theta^{*}=argmin(\ \|J\Delta\theta-\vec{e}\|^{2}+\lambda^{2}\|\Delta\theta\|^{2}) \quad (1)$$

$J(\theta)$ can be considered as a $3 \times n$ matrix (n represents the number of DOF for one finger). The Jacobian $J(\theta)=\partial S/\partial\theta$ is also computed. $\Delta\theta^{*}$ is added repeatedly to present θ up to that threshold distance to reach the value we have set. This distance is denoted as $\vec{e}$, the distance between a finger's tip and base. By control of λ, stability is regulated. We set the threshold distance as 0.1 and the damping ratio as 1000.

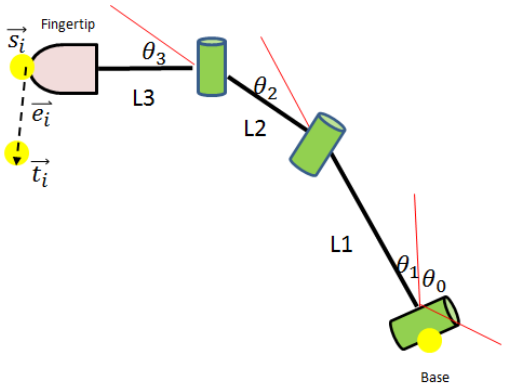

Fig. 5. Inverse kinematics for pose estimation

4 Implementation and Experiment

The proposed method runs on a computer equipped with i7 Core, 8GB RAM, GeForce GT 520M. The RGB-D camera is an Intel creative-gesture camera with a resolution of 640 × 480 for color images, and 320 × 240 for depth images. The HMD used is VUZIX 920AR, which has a resolution of 1024 × 768. To track the extrinsic parameter of a hand object, Sixense magnetic [15] is attached to the hand.

The base of the magnetic tracker is arranged on the reference coordinate of AR space, which quadrates the coordinate system. For the vision module, we use the openCV library, experiment with the IK module with the openGL library, and implement a demo with Unity 3D [16].

To enable a user to interact on a 3D virtual object with his own hand in AR space, a camera's extrinsic parameter is calculated and the local-reference coordinate is set in real space. According to this coordinate, the virtual object is registered in real space. This distinguishes the interface module for AR interaction from simply overlaying virtual objects in an image. Also a hand's extrinsic parameter is calculated, according to coordinates in real space. Figure 6 shows how coordinates interact with virtual objects. In other words, through the algorithm proposed in sections 3.1 and 3.2, this system estimates a hand's motion and enables a user to interact with virtual objects registered in real space.

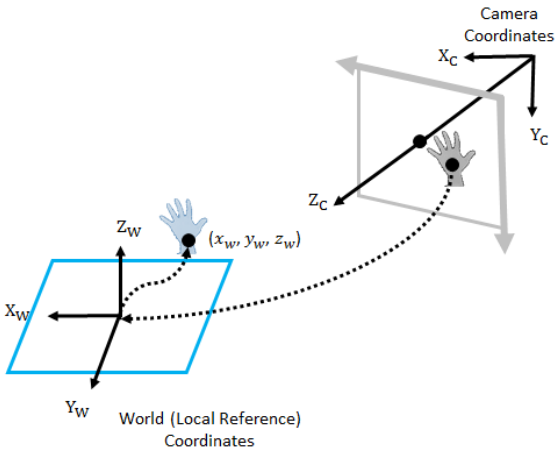

Fig. 6. Relative coordinate between a hand and real object

Two experiments compare our performance in detecting fingertips with Softkinetic's hand tracking module. Figure 7 show the performance, numbering finger detection in both the outside and inside of the palm. We configure our experiments, which involve bending and stretching fingers, in the regular sequence. Accurate detection of the number of fingers implies accurate detection of the fingertips' positions. The existing method showed some detection error outside of the palm and the method could not detect the finger inside of the palm properly. The method we propose, on the other hand, could detect fingertips robustly and stably. Figure 8 represents the quantitative experimental results of the inverse-kinematics algorithm applied to our system. It shows one fingertip's position in a virtual hand and the corresponding position of a fingertip of a real hand. The error boundary is 15mm (±5mm), including 15mm for camera error.

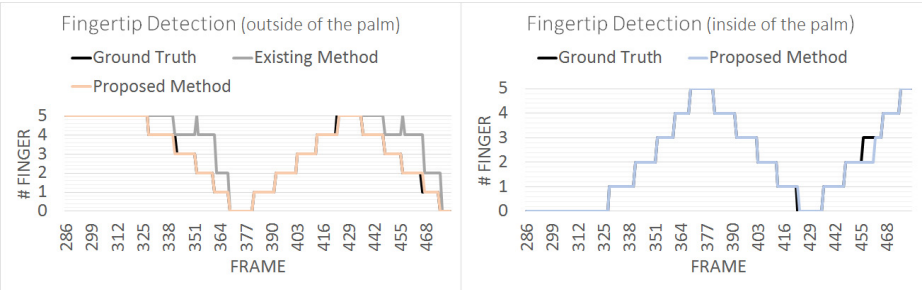

Fig. 7. Fingertip detection (Outside and inside of the palm)

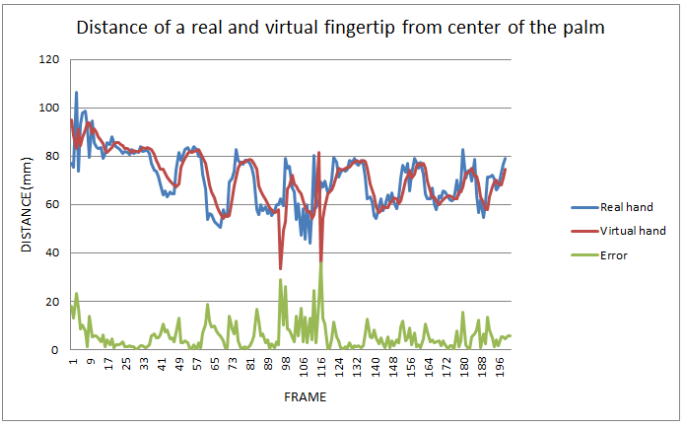

Fig. 8. Distance of a real and virtual fingertip from a palm center

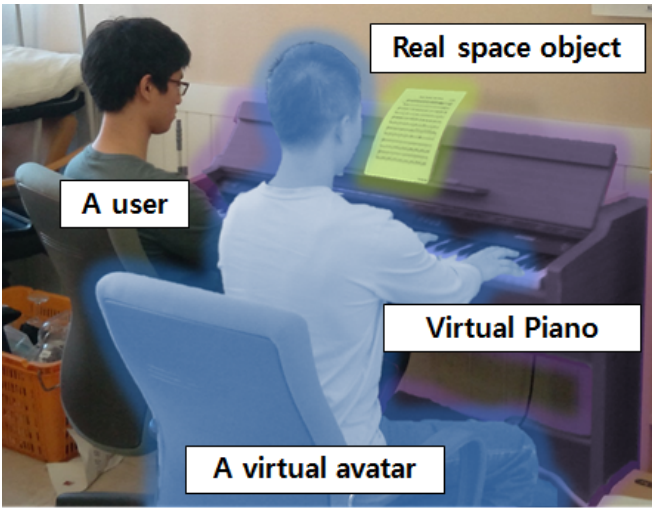

Fig. 9. Piano performance with virtual avatar

5 Conclusion and Future Work

This paper proposed a system for tracking hand motion with a near-range depth camera for augmented-reality interaction. Our system does not need special gloves, and estimates the pose of fingers in a wearable environment. After that, it helps a user wearing an HMD to interact with one's hand on virtual objects augmented from a real object of his interest. According to the miniaturization and weight-lightening of such devices as cameras and HMDs, this system will be used heavily in the wearable-computing environment for AR application.

This system does not consider the self-occlusion of the fingers that occurs when the camera view changes. In future works, we will make the hand-tracking system robust

to the self-occlusion of the fingers. It will estimate more poses of the fingers and have a greater number of interaction applications. Figure 9 shows our future scenario. A user can perform a virtual piano, registered on a physical table. The user can play the piano together with augmented virtual avatar. To realize this kind of application scenario, we will enable finger pose estimation under self-occlusion of the fingers.

Acknowledgment. This work was supported by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government (MSIP) (NRF-2010-0029751).

References

1. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3D tracking of hand articulations using Kinect. In: Proceedings of the 22nd British Machine Vision Conference, BMVC 2011, University of Dundee, UK, August 29-September 1 (2011)
2. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Tracking the articulated motion of two strongly interacting hands. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, Rhode Island, USA, June 18-20 (2012)
3. Wang, R.Y., Popovic, J.: Real-Time Hand-Tracking with a Color Glove. ACM Transaction on Graphics (SIGGRAPH 2009) 28(3) (August 2009)
4. Wang, R.Y., Paris, S., Popovic, J.: 6D Hands: Markerless Hand Tracking for Computer Aided Design. ACM User Interface Software and Technology, UIST (2011)
5. Coleca, F., Klement, S., Martinetz, T., Barth, E.: Real-time skeleton tracking for embedded systems. In: Proceedings SPIE, Mobile Computational Photography, vol. 8667D (2013)
6. LEAP MOTION, `https://www.leapmotion.com` (access date: February 5, 2014)
7. Wachs, J., Kölsch, M., Stern, H., Edan, Y.: Vision-Based Hand-Gesture Applications, Challenges and Innovations. Communications of the ACM (February 2011)
8. Lee, T., Höllerer, T.: Handy AR:Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking. In: Proceedings of the IEEE International Symposium on Werarable Computer(ISWC), Boston, MA (October 2007)
9. SOFT KINETIC, `http://www.softkinetic.com` (access date: February 5, 2014)
10. Albrecht, I., Haber, J., Seidel, H.-P.: Construction and Animation of Anatomically Based Human Hand Models. In: Eurographics Symposium on Computer Animation, p. 109. Eurographics Association (2003)
11. Ram Rajesh, J., Nagarjunan, D., Arunachalam, M., Aarthi, R.R.: Distance Transform Based Hand Gestures Recognition for Powerpoint Presentation Navigation. Advanced Computing 3(3), 41 (2012)
12. Wampler, C.W., Leifer, L.J.: Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators. Journal of Dynamic Systems, Measurement, and Control 110, 31–38 (1988)
13. Ha, T., Woo, W.: Bare hand interface for interaction in the video see-through HMD based wearable AR environment. In: Harper, R., Rauterberg, M., Combetto, M. (eds.) ICEC 2006. LNCS, vol. 4161, pp. 354–357. Springer, Heidelberg (2006)
14. Brunelli, R.: Template Matching Techniques in Computer Vision: Theory and Practice. Wiley
15. SIXENSE, `http://sixense.com/` (access date: February 5, 2014)
16. Unity3D, `http://unity3d.com` (access date: February 5, 2014)