INTERACTIVE MULTI-OBJECT TRACKING FOR VIRTUAL OBJECT MANIPULATION

Yibo Guo Michael Ying Yang * Bodo Rosenhahn

Institute for Information processing (TNT), Leibniz University Hanover
Appelstr. 9A, 30167 Hannover, Germany

Commission III/3

KEY WORDS: Kinect, multiple object tracking, gesture recognition, ICP, 1D signature

ABSTRACT:

We present an interactive system to manipulate a virtual object by tracking multiple hands in 3D space using a Kinect device. The system segments hand shapes from a captured 3D scene by using depth information and active contours. A hand shape is recognized by a trained naive Bayes classifier, whether it belongs to a palm, a pointing hand form or both hands with simple occlusion. A plane is approximated by using RANSAC for a palm hand form, while a vector from the hand centroid to the fingertip is obtained for a pointing hand form by using ICP as tracking method. A shape of simple occluded two hands is split into a palm and an incomplete pointing hand form, whose missing data is estimated by using PCA. The system works in semi-real time.

1 INTRODUCTION

Multi-Object tracking is an active and challenging research topic in computer vision (Smith et al., 2005; Berclaz et al., 2011). One of its instances, hand tracking in 3D space has been proposed in virtual reality applications including human-computer interaction, machine learning, human activity recognition, etc. The hand tracking problem in general has been proposed by many researchers using Kinect device. In (Raheja et al., 2011), distance transformation on depth images was used to detect fingertips and centers of palm. Frati and Prattichizzo (2011) analyzed convexity defect and bounding box of hand shape to obtain trajectories of fingertips. In (Oikonomidis et al., 2011), a 3D model of hand was used to compare the 3D hand pose in the scene and the status of each joint of the hand model was solved using a variant of Particle Swarm Optimization. This method was extended in (Oikonomidis et al., 2012) to solve the problem of tracking two interacting hands.

In this paper, we propose a human-computer interaction system to manipulate a virtual rigid object by 3D tracking multi-hand using a Kinect device. The system abstracts a plane for a palm hand form and a vector from the hand centroid to the fingertip for a pointing hand form. The plane and the vector are regarded as control data corresponding 6 and 5 degrees of freedom respectively for manipulation of the virtual rigid object. The system can also detect and handle the situation of simple occluded two hands. Applications of such system are numerous, including virtual grasp, CAD/CAM, virtual earth, and smart cities. The remainder of the paper is organized as follows. In Section 2, we present details of the proposed hand tracking system. Section 3 illustrates the results of extracted control data corresponding with predefined hand form according to the proposed approach. Section 4 concludes the paper and discusses the limitations and the future work.

2 PROPOSED FRAMEWORK

In this section, a hand tracking system is created to abstract the control data for manipulation of a virtual object. A Kinect for windows is used as RGB-D input device working at resolution 640×480 pixels and frame rate of 30 Hz. Due to the accuracy of Kinect depth data (Khoshelham and Elberink, 2012), the working range is restricted $0.5-0.8$ m along the depth dimension by using *near mode*.

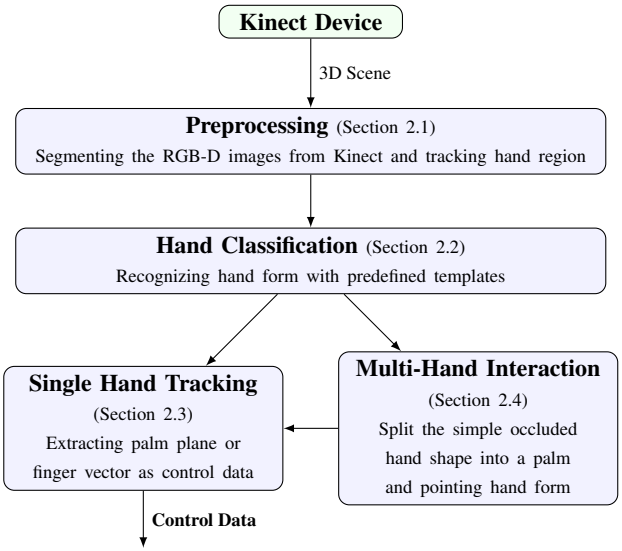


Figure 1: System overview

The system is composed of four major components, as illustrated in Figure 1. Each block in the diagram is a component of the system and is detailed in the corresponding sections. At first, the *preprocessing* component picks the hand shape by color and spatial information from the data, which are captured by a Kinect device. Then, the selected hand shape is identified through the *hand classification* component. If the given hand shape is palm or pointing hand form, the *single hand tracking* component tracks the position and orientation of the hand shape and estimates the plane of a palm or the vector of a pointing hand form as control data. If the given hand shape is simple occluded two hands, the *multi-hand interaction* component splits the shape into a palm and an incomplete pointing hand form, and then estimated missing data.

*Corresponding author. Email: yang@tnt.uni-hannover.de

2.1 Preprocessing

The purpose of the preprocessing component is using image segmentation methods of the fetched RGB-D image (see Figure 2a) from a Kinect device to track hand region in real time. By this process, a thresholding filter was used to remove the data which is out of the working range (see Figure 2b). Then the level set based active contour (Osher and Sethian, 1988) has been chosen to split the hand region by skin color within the rest of foreground.

(a) (b) (c)

Figure 2: Procedure of preprocessing. (a) A color image is aligned to the corresponding depth image. (b) The background of (a) is removed by using a thresholding filter. (c) Hand region is segmented by using FTC.

An approximation algorithm of the level set based curve evolution which are called fast two-cycle (FTC) Algorithm (Shi and Karl, 2008) was used for real-time implementation. The curve evolution is drove by the speed function which is based on the Chan-Vese model (Chan and Vese, 2001). A two-circle for the initialization of the FTC Algorithm is placed in or partially in the region of a hand. After curve evolutions, the segmentation results of the hand regions are obtained with both curves. The result curves can continue evolving with the next frame without reinitialization to track the hand movement. Figure 2c is shown the result of segmentation by using FTC.

2.2 Hand Classification

In order to generate control data of the manipulation of virtual object, hand forms are classified into *three* classes: (1) a hand with extended index finger, namely pointing hand form, (2) a palm, (3) simple occluded two hands. In a sample database, approximate 100 images per hand form are saved as training data. That is, approximate 300 training data in total. Figure 3 shows part of the samples.

The hand classification component recognizes the class of the given hand shape by using a naive Bayes classifier (NBC) (Caruana and Niculescu-Mizil, 2006), which trained by Hu-moments (Hu, 1962) of the hand region. The trained NBC is tested with approximate 3000 images and obtained a total recognition rate of 99.2%.

2.3 Single Hand Tracking

2.3.1 Tracking of Pointing Hand Form After the previous procedures, a hand region is segmented and recognized to the corresponding hand form. The control data of pointing hand form are defined by a vector from the centroid of hand to the index fingertip. In this subsection, the principle of iterative closest point (ICP) method (Horn, 1987; Zhang, 1994) is presented to track and match the fingertip.

The control data of pointing hand form are defined by a vector from the centroid B of hand to the index fingertip Q (see Figure 4a). In a local coordinate system of the pointing hand form,

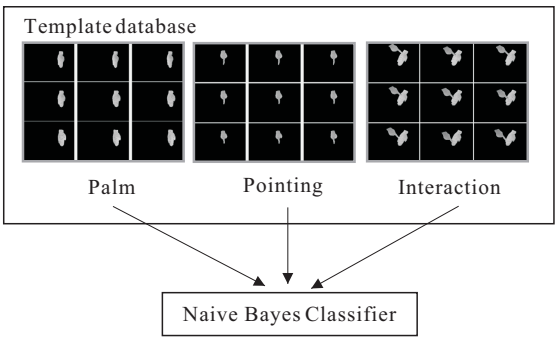
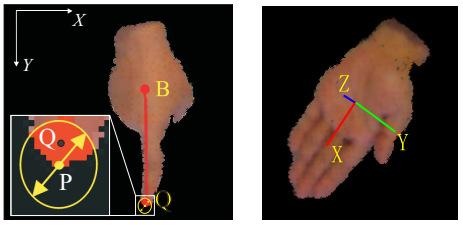


Figure 3: The NBC trained by Hu-moments of hand shapes



(a) Control vector (b) Palm plane

Figure 4: Control data definition

the position of index fingertip Q is defined as a mean value of points, which are surrounding the point P with the maximal Y coordinate. The radius of the surrounding region is 5 mm.

To obtain the correct position of fingertip, a given pointing hand form must be transformed to the local coordinate system. By using ICP algorithm, the transformation $\mathcal{T}$ that best aligns the point cloud of a given hand form to a mean template is found. All the points of the given point cloud are transformed into a local coordinate system defined by the mean template. The position of the fingertip Q and the hand shape center B are calculated according to the definition in the local coordinate system. By using the inverse transformation $\mathcal{T}'$, the local coordinate system of the control vector $\overrightarrow{BQ}$ are converted into the world coordinate system.

Mean Template Generation To obtain a hand shape defined in local coordinate system, a mean template is generated by a set of hand form samples from a database. The generation procedure is illustrated in Figure 5 and interpreted by following steps:

1. **Sample alignment** To generate a mean template, a set of pointing hand form samples are saved in the database (see Figure 5(A)). Through the hand segmentation component, the contours of samples are extracted. A contour whose index fingertip keeps in the front of hand shape with maximal Y coordinate is selected as model for registration of ICP. The coordinate system of the model is regarded as local coordinate of the hand form. Other contours are aligned with this selected contour by using ICP and then transformed to the local coordinate system.

2. **1D signature computation** By computing distances from points on the aligned contour to the centroid $B_{x,y}$, a *centroid distance* is calculated as a 1D signature of the contour. To back convert to the contour from the 1D signature, the angle of each point is also computed. The distances and angles are described in middle of Figure 5(B). The black curve is according to the radius while the color indicates the angles of the points on the contour. The distance $dist$ and angle

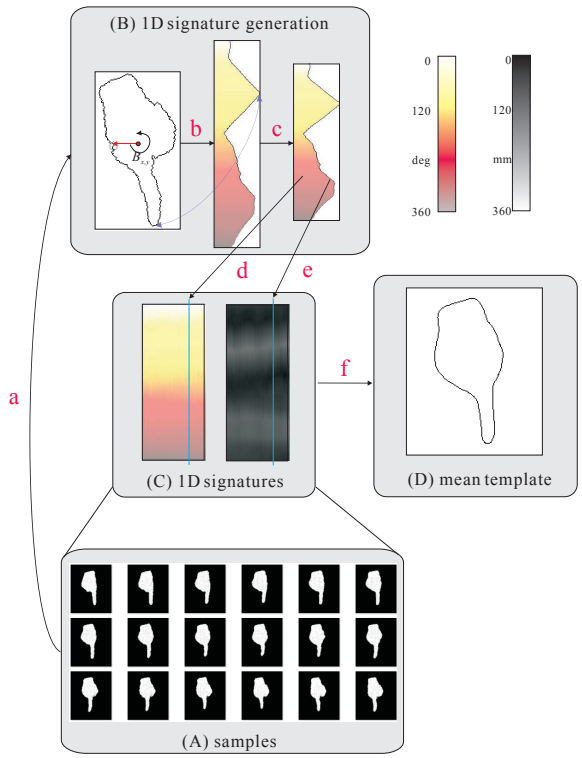


Figure 5: Mean template generation.

ang between a point $G_{x,y}$ on the contour and the centroid $B_{x,y}$ can be computed with follow equations:

$$dist = \sqrt{(G_x - B_x)^2 + (G_y - B_y)^2} \quad (1)$$

$$ang = \arctan(G_y - B_y, G_x - B_x) + \pi \quad (2)$$

3. **Mean template generation** Due to the different perimeters of contours, 1D signatures of samples are resampled with a fixed point count (right Figure 5(B)). The resampled 1D signatures are saved in radius and angle matrices (Figure 5(C)). Each column of the matrices corresponding to a contour of the hand shape. An average 1D signature is computed from both matrices and converted to a mean contour of the set of samples (Figure 5(D)).

Hand Shape Alignment A given hand shape contour in the current frame is aligned to the mean hand shape contour by using ICP. Due to the movement of the hand continuous, the transformation results of ICP are saved and regard as initial transformation for the contour in next frame. Therefore the speed of convergence of the ICP alignment can be accelerated in the next frame.

Control Vector Generation The given hand shape is converted into the local coordinate system according to the transformation result of the ICP. To find out the fingertip, the point with maximum Y coordinate, point P, is selected (see Figure 4a). Then a mean value of points of the finger which are surround the point P is calculated as the position of the fingertip Q. The centroid of the hand shape B is approximated with the mean position of points on the hand contour. The control vector $\overrightarrow{BQ}$ are back projected in spatial space in the world coordinate system.

2.3.2 Tracking of Palm Hand Form

(a) Point cloud (b) voxel size=15mm

Figure 6: Reduction of points quantity by using a voxel grid filter

Palm plane approximation To estimate the parameters of the plane, following processes are implemented:

1. A hand shape recognized as palm is converted to a point cloud in 3D.

2. The point cloud is down-sampled by a voxel grid filter to reduce point quantity. The voxel grid filter creates a 3D voxel grid over the point cloud. In each voxel, all the points are approximated with their centroid. The palm surface will be represented more accurately by the down-sampled point cloud. As the accuracy of the working range is about 3 mm, the voxel size is at least set to 3 mm to guarantee stability. Figure 6 shows the original point cloud and the filtered point cloud by a voxel grid filter with voxel size 3 mm. On the other hand, the plane of hand palm is also not ideal. The altitude difference between raised parts and the valleys of the palm is about 10-15 mm. Therefore, the voxel size is set to 15 mm for the plane estimation.

3. Parameters of a plane equation approximated by the down-sampled palm point cloud are estimated through the random sample consensus (RANSAC) method (Fischler and Bolles, 1981). The plane equation is used in Hessian normal form,

$$\overrightarrow{\mathcal{N}} \cdot x = -c, \quad (3)$$

where c is the distance of the plane from the origin, $\overrightarrow{\mathcal{N}}$ is the normal vector of the plane.

4. A parameter vector of a good plane model is approximated by the RANSAC algorithm. Red points in Figure 7 are inliers of the plane model. The normal vector of the palm centroid is calculated as the control data and is presented by a blue line.

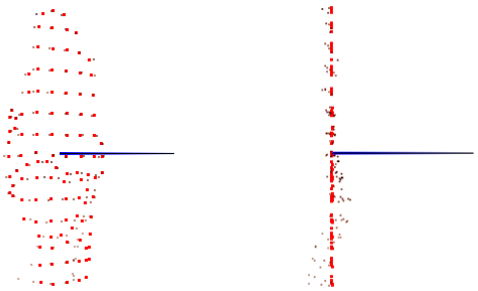


Figure 7: Approximation of a palm plane from a point cloud. Red points are inliers of the approximated plane. The blue line is the normal vector of the plane.

2.3.3 Control Data Smoothing By applying the previous procedures, control data of a pointing or a palm hand form are obtained from captured frames. The control data sequence generates trajectories in 3D space. Due to the sampling frequency and the measurement error, the control data trajectories are not as smooth as the ground truth. In order to reduce the influence of errors, a SavitzkyGolay smoothing filter (Savitzky and Golay, 1964) is used to determine the smoothed value for each point on a trajectory.

2.4 Multi-Hand Interaction

The single hand tracking component has the ability to detect and track multi-hand at a same time, if the hands do not occlude or contact between each other. The main task of the multi-hand tracking component is to analyze a situation of simple occluded two hands. Simple occluded two hands, here refers to a pointing hand form over a palm. To obtain the control vector of a pointing hand form, the fingertip in the overlapping region must be recognized and tracked. However, due to the skin colors of both hands are very similar, it is difficult to split the contours of both hands in the overlapping area by using the active contour method described in Section 2.1.

For the purpose of estimating the hand form contour in the overlapping area, the overlapping hands contour is split into a palm and a missing data pointing hand form according to a mask which is generated by template samples of the palm hand form. Then the missing points of the overlapping region is estimated by using principal component analysis (PCA) of pointing hand form template samples from the database.

2.4.1 Split of occluded Hand Shape Once the system detects the occlusion of two hands, the given shape should be split into palm and pointing hand forms. Then the 1D signature of the incomplete contour of pointing hand form is estimated. Figure 8 presents this procedure by following steps:

1. A contour of the given interaction shape is extracted (see Figure 8(a)) and then aligned to a mean template of the palm hand form (see Figure 8(b)). The result of alignment is presented in Figure 8(c).

2. A mask image of the palm hand form is used to split the result contour from step 1 into a palm (see Figure 8(f)) and an incomplete pointing hand form 8(e)). Each point of the contour is determined by the corresponding coordinate in a mask image, whether it belongs to a palm or pointing hand form. The mask image of the palm hand form is generated by palm template samples from the database. Contours of all the palm template samples are extracted and points of contours are transformed into a polar coordinate system. Then each point of the contour is expressed by a radius and an angle. For each angle of all the points in the polar coordinate system, the min- and maximum radius are estimated. The points between min and maximum of this angle are labeled as masks (see Figure 8(d) left). The points which labeled as masks are transformed back to the Cartesian coordinate system and generate a mask image of the palm hand form (see Figure 8(d) right).

3. The contour of the pointing hand form (see Figure 8(e)) is aligned to the mean template of the pointing hand form (see Figure 8(g)). The result is shown in Figure 8(h)).

4. The 1D signature of the result of step 1 is computed (see Figure 8(i)).

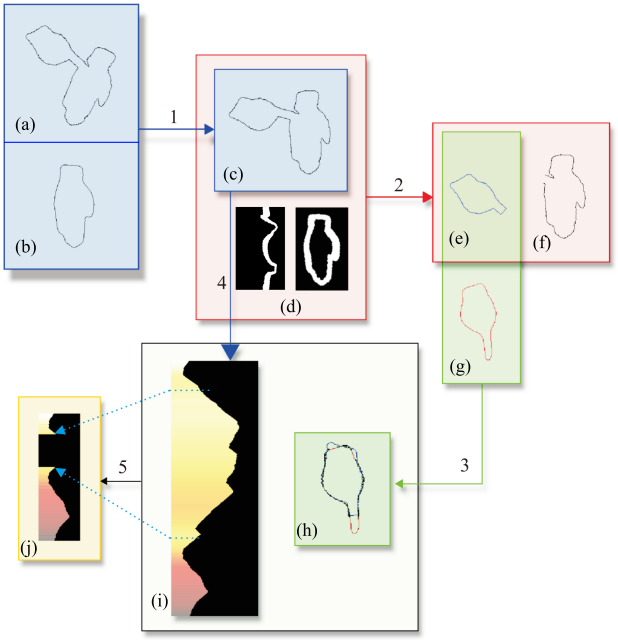


Figure 8: Obtaining an incomplete 1D signature of a pointing hand form from a given simple occluded hand shapes. In step 1, a contour (a) of the given interaction shape is extracted and then aligned to a mean template of palm hand form (b). The result of alignment is presented in (c). In step 2, a mask image of the palm hand form (right side of d) is generated by palm template samples, which are transformed into a polar coordinate system (left side of d). In step 3, the contour of the pointing hand form (e) is aligned to the mean template of the pointing hand form (g). The result is shown in (h). In step 4, the 1D signature (i) of the result of step 1 is computed. In step 5, an incomplete 1D signature of the pointing hand form contour (j) is estimated and resampled by according to the results of step 3 and 4.

5. According to the results of step 3 and 4, each point of the 1D signature is detected whether it belongs to the pointing hand form. Then an incomplete 1D signature of the pointing hand form contour is estimated and resampled (see Figure 8(j)).

2.4.2 Incomplete Data Estimation To estimate the missing data of pointing hand form, 1D signatures of template samples of pointing hand form in the database are analyzed by PCA (Turk and Pentland, 1991). Each 1D signature has 360 points that are represented as the angle and distance that is total 720 dimensions. The angles and distances are normalized. Figure 9(a) shows the 1D signatures matrices of angles (the color image on the left) and distances (the black and white image on the right) from 74 pointing hand form samples. Each column of the matrices corresponding to a template sample.

The 720×74 1D signature matrix is analyzed by PCA. A mean signature and a covariance matrix are computed. The eigenvalues and eigenvectors of the covariance matrix are obtained by using singular value decomposition (SVD). Five eigenvectors with maximal weight are selected as feature vectors to reduce the data dimensions. The five eigenvectors provide more than 90% information.

Figure 9(c) represents a few states of 1D signature in polar coordinate system at different iterations. The red curve at 0 iteration is the mean 1D signature. After 50 iterations, the missing values are estimated and remained stable. The estimated 1D signature is then converted into a pointing hand form (see Figure 9(d)).

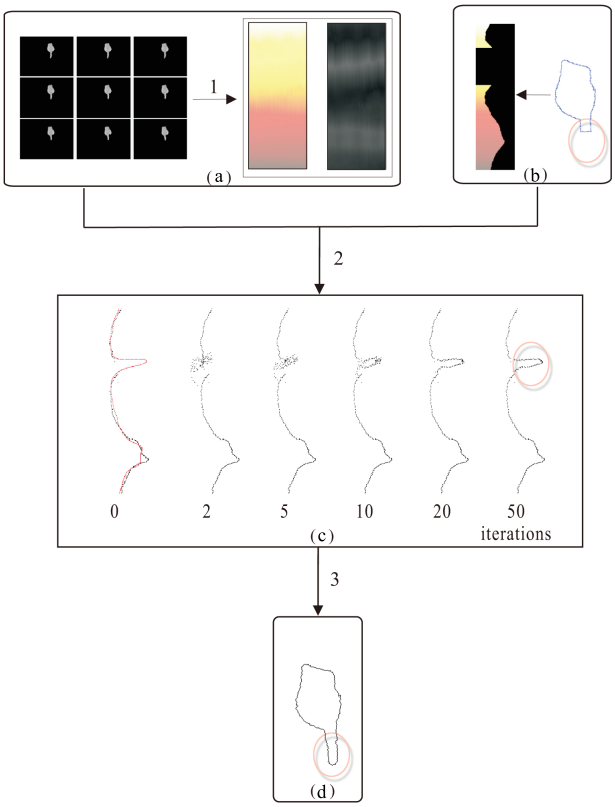


Figure 9: Estimation of missing data by using PCA. (a) shows the 1D signatures matrices of angles (color image on the left) and distances (black and white image on the right) from 74 pointing hand form samples. Each column of the matrices corresponding to a template sample (see Section 2.3.1). (b) is a given incomplete 1D signature which is obtained in Section 2.4.1. (c) presents the iteration process of PCA to estimation the missing data of 1D signature of an incomplete contour. (d) illustrates the result of the estimation of the missing data.

3 EXPERIMENTAL RESULTS

In this section, the results of extracted control data of a hand form are illustrated according to the proposed approach. The experiments are designed to estimate the accuracy and stability of the control data.

To estimate the accuracy of the control data, some trick scenarios are analyzed. In these scenarios, for example Figure 10, a pencil is hidden under the index finger and draws a trajectory of the fingertip on a paper. The trajectory on the paper is scanned as "ground truth" data (see Figure 10b) to compare with the synchronized control data generated by the system. Figure 10c shows the smoothed trajectory of the fingertip. The result compared to ground truth data are represented in Figure 10d, which shows that there is only minor difference between smoothed trajectory and scanned trajectory. Figure 10e and 10f show the 3D demonstrations of the fingertip trajectory (blue) and the hand centroid trajectory (orange).

Figure 11a represents a scenario of a waving palm. The tracking result is shown in Figure 11b. The red points of the hand belong to a palm plane. The orientation of the palm is the vector from palm centroid (green) to fingertip of middle finger (red). The normal vector of the palm plane is through the centroid with 10 mm length (from green to cyan).

Figure 12 illustrates trajectories of two hand interaction. The red curve is the trajectory of the index fingertip, which is the results of estimating missing points of the pointing hand form contour. The green curve is the trajectory of the centroid of the hand.

Table 1 presents the average computation time of major algorithms or components. The system without multi-hand interaction component works in real time and the whole system works in semi-real time.

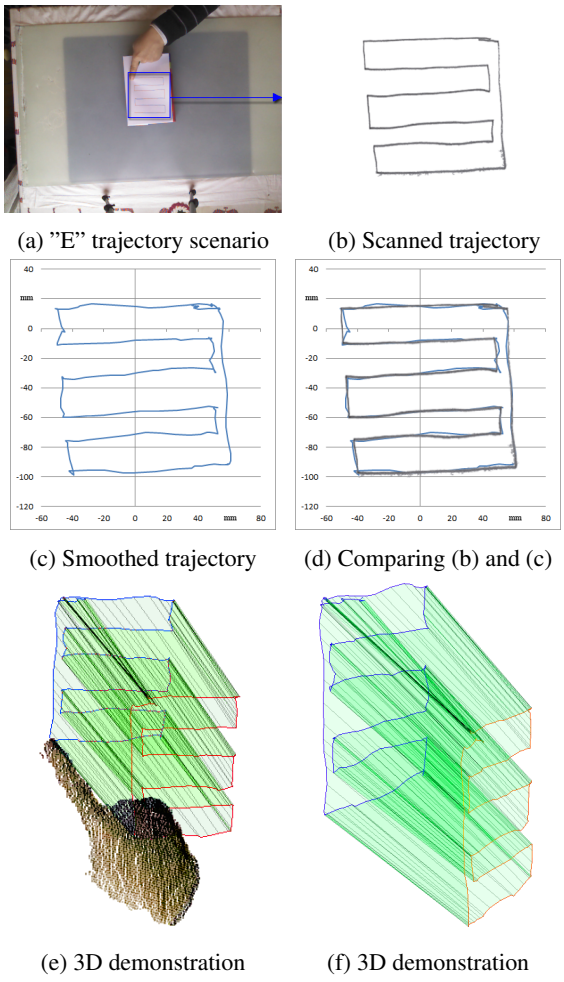


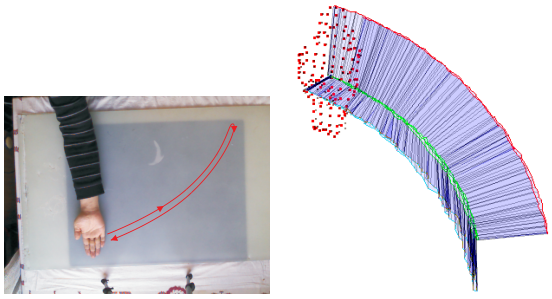
Figure 10: Tracking a pointing hand form (171 frames)

Table 1: Computation time of major algorithms or components

	Section	Time (ms)
Preprocessing	2.1	13
Hand classification	2.2	4
ICP	2.3.1	18
1D signature	2.3.1	1
RANSAC	2.3.2	6
PCA missing data estimation	2.4.2	36

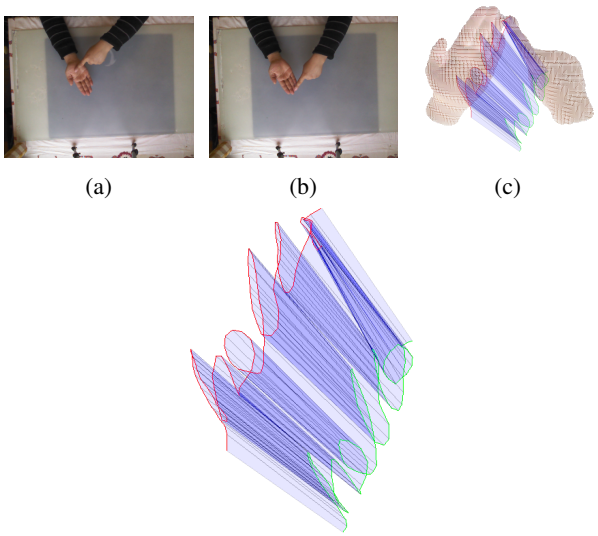
4 CONCLUSIONS AND FUTURE WORK

In this paper, a system is developed by using a Kinect device to track the movement of hands in spatial space and extract the necessary 3D control data. By using depth information and the level set based active contour method, the hand shapes are segmented.



(a) (b)

Figure 11: Tracking a palm (152 frames). (a) is a waving scenario of a palm. (b) is the 3D demonstration of the palm cloud, trajectory of a control vector from palm centroid (green) to fingertip (red), and the normal vector of the palm plane through the centroid with 10 mm length (from green to cyan).



(a) (b) (c) (d)

Figure 12: Trajectories of tracking multi-hand interaction. (a) and (b) show the scenario of multi-hand interaction with occlusion. (c) and (d) show result of the control vector from hand centroid (green) to index fingertip (red). The index fingertip in the occluded region is reconstructed by using the PCA method.

A NBC is used to classify a given hand shape, whether it belongs to a palm, a pointing hand form or both hands with simple occlusion. If a given hand shape is a palm, a plane of the palm is approximated by using the RANSAC method. If a hand shape is a pointing hand form, a vector from the hand centroid to the fingertip is obtained by using the ICP method. If the system detects that the given shape contains both hands with a simple occlusion, the shape is split into a palm and an incomplete pointing hand form. Missing data of the incomplete pointing hand form is estimated by using the PCA method. Then a pointing hand form is reconstructed and the control vector is extracted.

This system has been made in robust hand gesture interaction. However, more hand forms can be imported into the system to meet the needs of more applications. The accuracy of the data should be improved by using a next generation Kinect device or using multi-Kinect devices. The system can be speed up by using parallel programming to achieve the needs of real-time computing.

References

Berclaz, J., Fleuret, F., Türetken, E. and Fua, P., 2011. Multiple object tracking using K-shortest paths optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 33, pp. 1806–1819.

Caruana, R. and Niculescu-Mizil, A., 2006. An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning, ICML '06, pp. 161–168.

Chan, T. and Vese, L., 2001. Active contours without edges. IEEE Transactions on Image Processing Vol. 10, pp. 266 –277.

Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM Vol. 24, pp. 381–395.

Frati, V. and Prattichizzo, D., 2011. Using Kinect for hand tracking and rendering in wearable haptics. In: IEEE World Haptics Conference (WHC), pp. 317–321.

Horn, B. K. P., 1987. Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America Vol. 4, pp. 629–642.

Hu, M.-K., 1962. Visual pattern recognition by moment invariants. IRE Transactions on Information Theory Vol. 8, pp. 179–187.

Khoshelham, K. and Elberink, S. O., 2012. Accuracy and resolution of Kinect depth data for indoor mapping applications. Sensors Vol. 12, pp. 1437–1454.

Oikonomidis, I., Kyriazis, N. and Argyros, A., 2011. Efficient model-based 3D tracking of hand articulations using Kinect. In: Proceedings of the British Machine Vision Conference, pp. 101.1–101.11.

Oikonomidis, I., Kyriazis, N. and Argyros, A., 2012. Tracking the articulated motion of two strongly interacting hands. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1862–1869.

Osher, S. and Sethian, J. A., 1988. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics Vol. 79, pp. 12–49.

Raheja, J., Chaudhary, A. and Singal, K., 2011. Tracking of fingertips and centers of palm using Kinect. In: Third International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM), pp. 248–252.

Savitzky, A. and Golay, M. J. E., 1964. Smoothing and differentiation of data by simplified least squares procedures. Analytical Chemistry Vol. 36, pp. 1627–1639.

Shi, Y. and Karl, W., 2008. A real-time algorithm for the approximation of level-set-based curve evolution. IEEE Transactions on Image Processing Vol. 17, pp. 645–656.

Smith, K., Gatica-perez, D., marc Odobez, J. and Ba, S., 2005. Evaluating multi-object tracking. In: Workshop on Empirical Evaluation Methods in Computer Vision.

Turk, M. and Pentland, A., 1991. Eigenfaces for recognition. Journal Cognitive Neuroscience Vol. 3, pp. 71–86.

Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision Vol. 13, pp. 119–152.