A Novel Approach to Extract Hand Gesture Feature in Depth Images

Zhaojie Ju, Dongxu Gao, Jiangtao Cao, Honghai Liu

Abstract This paper proposes a novel approach to extract human hand gesture features in real-time from RGB-D images based on the earth mover's distance and Lasso algorithms. Firstly, hand gestures with hand edge contour are segmented using a contour length information based de-noise method. A modified finger earth mover's distance algorithm is then employed applied to locate the palm image and extract fingertip features. Lastly and more importantly, a Lasso algorithm is proposed to effectively and efficiently extract the fingertip feature from a hand contour curve. Experimental results demonstrate performance of the proposed approach.

Keywords Hand gestures, Feature extraction, Kinect sensor, EMD

Z. Ju
University of Portsmouth, Portsmouth, UK
State Key Laboratory of Mechanical System and Vibration, China
e-mail: zhaojie.ju@port.ac.uk
D. Gao
University of Portsmouth, Portsmouth, UK
e-mail:dongxu.gao@port.ac.uk
J. Cao
Liaoning Shihua University, Fushun, China
e-mail:cigroup@123.com
H. Liu (Corresponding Author)
University of Portsmouth, Portsmouth, UK
e-mail: Honghai.liu@port.ac.uk

1. Introduction

Kinect is Microsoft's somatic human-computer interaction device developed for its game console XBOX360. Actually it is a three-dimension camera that can provide color image stream, depth image stream, skeleton image stream and voice stream in real time. Thus it is able to catch the movement of players in three-dimension. Then we can use simple gestures and voice to do the human-computer interaction. Kinect hardware includes a color camera, a depth camera witch consist of an infrared emitter and a infrared receiver, a set of microphone sensors, a acceleration sensor and a servo motor that drive the pitching action.

Traditional three-dimension measurement use structured light or time of flight (ToF) [1]. The illumination is dispersing in time and space. While Kinect depth camera is based on a new technology called Light-coding, in which continuous illumination is used. In this way, general CMOS sensor can be used to reduce cost. In addition, being a game peripheral, Kinect has strong real-time computing power, especially for its skeleton stream witch is

calculated using depth information. Due to these advantages, Kinect sensor works as a low-cost and high-performance hardware platform in many research field besides computer games, such as virtual trying on clothes [2], three-dimension scanner [3], robot navigation [4] and so on [5].

Kinect is designed as a game peripheral, so despite its powerful real-time human-computer interaction function, there are some weaknesses in respects that is of no importance for game user experience due to cost cutting. And these become obstacles for the application of Kinect in scientific research. Specifically, the resolution of Kinect's color camera and depth camera too low to recognise some details in particular scene; the color image data and depth image data cannot match with each other ideally, so calibration is necessary [6]; depth image data and skeleton data are usually mixed up with lots of noise, because they are created by embedded computer using its inner algorithm, and this will be a problem for some high-robust-needed applications; the depth data has a limitation of distance measurement which is designed for computer game players, Kinect cannot provide reliable distance information exceed this limitation [7]. So many projects using Kinect as a hardware platform are trying to find the way to make up these weaknesses. They include the calibration of the Kinect's color image data and depth image data [8]; algorithm for calculation of skeleton data using raw data [9]; some filtering algorithm of Kinect data streams [10], and so on.

Hand is the most flexible connector between human and the outside world, so do human-computer interaction based on physical sensory. The research of hand gesture recognition based on the Kinect has become very important and also has gained widely attention. To recognize hand gesture, there are three aspects of important work that need be discussed. First of all, hand detection is the basis of all other analysis algorithm. Hand detection base on two kinds of data, they are RGB pixels and depth data. For hand detection from depth data, segmentation is considered as a depth clustering problem, where the pixels are divided into different depth levels in [11]. The critical part is to determine a threshold, indicating at which depth level the hand is located. In [12], Lianget comes up with a concise method, using a clustering algorithm followed by morphological constraints, he detect hands on the depth images, then a distance transform to the segmented hand contour is performed to estimate the palm and its center. Secondly, hand pose estimation is the key of most hand gesture recognition research. Finger-Earth Mover's distance is proposed in [13] to measure the dissimilarities between different hand shapes. Skeleton-based approaches deduce the hand pose based on the configuration of the hand skeleton. The skeleton generation is the key to these methods. A pioneering work [14] is the random decision forests (RDF)-based hand skeleton tracking, which can be seen as a hand-specific version of Shotton's human-body skeleton tracking. It performs per-pixel classification by means of the RDF technique and assigns each pixel a hand part. Then, the mean shift algorithm is applied to estimate the centers of hand parts to form a hand skeleton. Finally, hand gesture classification makes it possible to understand the meanings of hand gestures [15]. Tang et al. [16] establish a real-time system that is able to identify simple hand gestures such as grasping and pointing. In this system, a person's hand is estimated based on a skeletal tracker. Next, local shape-related features, such as radial histogram and modified speeded-up robust features (SURF) [17], are extracted from the image. A support vector machine (SVM) classifier (with a radial basis function) is used to distinguish the hand gestures. In this paper, a real-time hand gesture feature extraction method is proposed by a novel finger earth mover's distance algorithm and Lasso algorithm to locate the palm and effectively extract the hand contour curve. The diagram of the proposed approach is shown in Fig. 1.

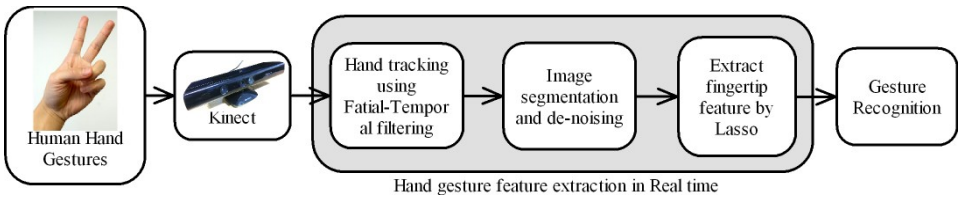

Figure 1: The proposed approach for hand gesture feature extraction in real time

2. Image segmentation and de-noising combining with depth information

2.1 Image segmentation

Kinect sensor [18] can output color information and depth information flow, and with the combination of the two information flows, the required palm image can be drawn out from complex background. Color information flow can be set with several grades of resolution ratio and different formats. High-resolution ratio can provide more information, while the update rate is low. It is opposite to the low-resolution ratio, users can select according to specific application. To provide depth information ratio is one of the characteristics of Kinect, and is also the foundation of the realization of basic physical sensory. The value of each frame of image pixel of depth information flow represents the minimum distance between the point on the image and the real object. Then it is possible to track the action of human or erase the background object by the depth information.

Besides, Kinect can also provide the inner processor processed and meaningful information for users, such as player-separated data. Separated data is provided with bitmap format, each value of image pixel is corresponding to the number of the player who is the nearest to the camera at the position of the pixel in the field of view.

The palm will be roughly segmented based only on the depth information. Firstly, Spatial-temporal filtering (STF) [2, 3] is employed to track the hand position and based on the tracking result the hand initial depth can be automatically chosen, as shown in Fig. 2. Then, the initial depth is used to determine the depth threshold value, based on which the palm images are segmented. In addition, the initial edge of the hand gesture can be achieved using Sobel method [4].

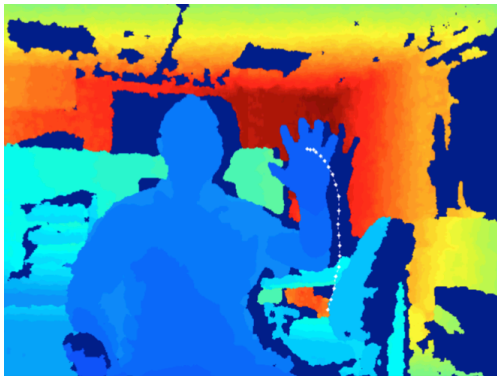

Fig. 1. Hand tracking using Spatical-Temporal filtering [2,3],and the hand trajectory is shown in white dots

2.2 Select the edge contour of image

With the above image segment result, it is much simple to select the edge contour of image. Traversal the pixel of four directions can get the contour of all eight directions. The pseudo-code shall be found in the algorithm 1 in the appendix.

2.3 De-noise by contour length information

The palm image information extracted by method above is of severe noise. Because the depth information provided by Kinect sensor has a low-resolution ratio. With such ratio it is difficult to recognize the object like palm, which is with abundant details. Also, the continuity of player separation data can not be fully ensured among the frames, it is with some breaks in times. All the above reasons make the palm image segmented by only Kinect depth information very hard and player separation data is not used in recognition algorithm, which is of high robustness. In figure 2, there are some holes in the segmented palm contour. These holes are wrongly taken as pixel that is not belonging to the palm.

Fig. 2. Problems of noise of image segmentation

By experiment, there are mainly 2 kinds of noises caused by image segmentation: one kind of noisy point is connected with the background, because of its existence, the palm image is separated into several parts, such as the No.3 black part in figure 2; another kind of noisy point is inside the palm image, around them, are the recognized palm image pixel. The existence of such noise will lead to the whole structure. Such as No.1 and 2 in figure 2.

In order to filtrate the first kind of noise, we use closed operation that is widely used in image morphology, and firstly expansion and then corrosion should be undertaken in figure 3. The principle shall be found as follows:

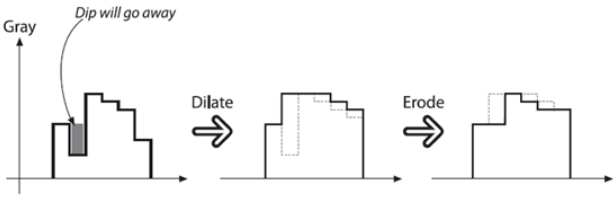

Fig. 3. Closed operation of morphology

The operation of morphology can also filtrate the second kind of noise, but we consider a method that can totally eliminate the second noise. It is a method that can use the contour length information. The principle of the method is: Suppose the shape of palm and the noise hollow are all the edges, extract the respective contour point, divide the eight connected

contour point as a group, you will get several groups of contour chain. Calculate the length of each group of contour chain, and find out the longest group. The noise hole is relatively smaller compared with the real size of palm, so we can consider the longest group of contour chain as the real contour of palm; other short chains are formed because of the second kind of noise. Figure 4 indicates that such method can effectively recognize and filtrate the second kind of noise.

Fig. 4. Use long contour information to filtrate the noise

3. Extract fingertip feature by Lasso

In this section, the earth mover's distance is introduced and modified, based on which the Lasso algorithm is proposed.

3.1 Earth Mover's Distance

Earth Mover's Distance (EMD) is defined as measure criterion compared with histogram. The problem it concerned is how to change the histogram from one shape into another one by certain movement (including move part of the histogram or whole to a new position). Generally, people always construct the information of histogram by statistical methods with the application of EMD, then match the different histograms of different images according to the algorithm to distinguish different images. Zhou Ren raised a new method to construct histogram, which is FEMD (Finger Earth Mover's Distance [6]). In this method, it is the direct conversion from gesture contour into histogram instead of the image calculation histogram information applied in EMD calculation. The detailed steps of the construction of finger contour histogram:

A. Segment the complete image of palm

B. Find out the central point of palm image by certain method

C. Start from the left contour of wrist; traverse the whole contour of palm until the right of wrist. Record the distance from the central point to each contour point and the angel from the start around the palm.

D. Suppose the angle as X-coordinate and the distance data as Y-coordinate, form the curve graph.

Find the corresponding position of each finger in curve graph, construct the histogram data required by EMD algorithm with the area by its width and the X-coordinate, as shown in figure 5.

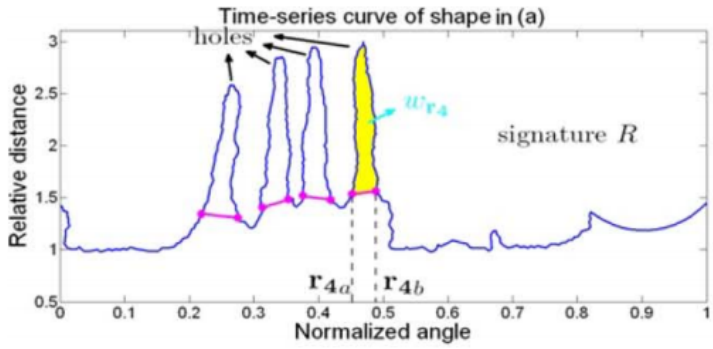

Fig. 5. FEDM gesture recognition algorithm

r_{4a} and r_{4b} are the two identified points for the fourth finger; holes mean the identified peaks. To compute the FEMD, we need to compute the value of F. F is defined by minimizing the work needed to move all the earth piles:

$$F = \arg\min WORK(R, T, F) = \arg\min \sum_{i=1}^{\bar{m}} \sum_{j=1}^{\bar{n}} d_{ij} f_{ij},$$

$$s.t. \begin{cases} f_{ij} \geq 0 \cdots\cdots\cdots\cdots 1 \leq i \leq \bar{m}, 1 \leq j \leq \bar{n}, \\ \sum_{j=1}^{\bar{n}} f_{ij} \leq \omega_{r_i} \cdots\cdots\cdots 1 \leq i \leq \bar{m}, \\ \sum_{i=1}^{\bar{m}} f_{ij} \leq \omega_{t_j} \cdots\cdots\cdots 1 \leq j \leq \bar{n}, \\ \sum_{i=1}^{\bar{m}} \sum_{j=1}^{\bar{n}} f_{ij} = \min(\sum_{i=1}^{\bar{m}} \omega_{r_i}, \sum_{j=1}^{\bar{n}} \omega_{t_j}) \end{cases} \quad (1)$$

Matric f is the flow matrix in EMD, matric d is the ground distance matric of the tow signature, R and T. We find the minimum work needed to move the earth piles. The first constraint restricts the moving flow to one direction: from earth piles to the holes. The last constraint forces the maximum amount of earth possible to be moved.

3.2 Lasso Algorithm

In order to use FEMD algorithm, the position coordinates of palm need to be determined. One method is to calculate the center of gravity position of the whole palm as the palm required by the FEMD algorithm. Such method is of clearly thought and simple algorithm, but there is an obvious disadvantage: theoretically, the palm center in the FEMD algorithm shall be the center of palm center image not including the finger part. Calculating the center position of the whole image without any distinction will definitely lead into deviation. Another simple and available method is applied in this paper, which is to find out all small regions nearest to the edge contour point by image corrosion. Then the position is the center of region similar with the palm center. The last rest region is single pixel, and we can get the center of circle for palm contour image by such method under the most ideal circumstance. Such method can also judge the threshold value by setting the size of corrosion and adjusting the area of the rest region, also limited the deviation into a reasonable range. The program flow chart shall be found as figure 6. There are two loops: one is for the condition of square and the other is

for the condition of the threshold. After processing, change of the algorithm is correct and available, the robustness and instantaneity are perfect.

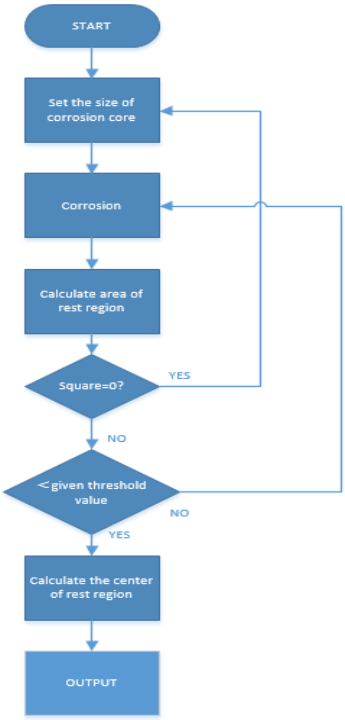

Fig. 6. Palm extract algorithm

Besides the calculation of palm center, the extraction of fingertip feature is another important step to construct palm contour curve by EMD algorithm histogram. The fingertip feature required by FEMD algorithm consists of two kinds of data, which are the X-coordinate of finger root and the areas between the fingers. In [6], two kinds of method for extraction of features of fingertip was mentioned, threshold and Near-Convex Decomposition. Near-Convex Decomposition can segment the feature of fingertip in palm image accurately, and is with well robustness. The disadvantage is also obvious, such as complicated algorithm and awful real-time. The threshold applies the simplest strategy, by given a certain value of threshold and all parts that higher than the threshold value shall be considered as the feature of fingertip. This method has high requirement of palm contour curve, and easily to be effected by noise.

In our paper, another new algorithm to extract the features of fingertips, called Lasso, is proposed with both fast computation and high robustness. By imitating the looping into the pillar, take the fingertip as a pillar. The width of these pillars is almost the same. Extraction of the features of these pillars is like the process of looping the pillars with the lasso which diameter is similar with the width of the pillars. When the lasso goes down to the bottom or it is tightened, the extraction can be seen as finished. The process of the above algorithm shall be found in the following figure 7. The algorithm of the Lasso can be found in algorithm 2 in the appendix.

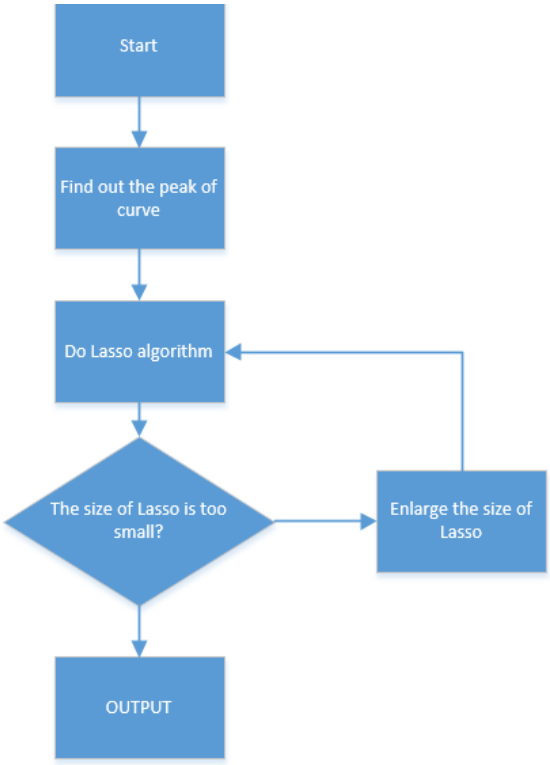

Fig. 7. Process of extract of fingertip features

Compared with Near-Convex Decomposition algorithm, Lasso algorithm deal with a simple two-dimensional curve instead of three-dimensional image. And for each data point of the curve, it needs only one traverse in the worst case. In fact, those data points that do not belong to a fingertip feature will be ignored automatically. So the lasso algorithm has a very low time complexity and shows outstanding performance as a real-time algorithm. Also, lasso algorithm shows pretty good reliability. There is significant difference between fingertip features and noise signal from the perspective of lasso principle. And lasso algorithm support for dynamic modification of lasso length, so that fingertip features with different size can be adapted. Several examples results achieved by lasso algorithm to extract the fingertip features are shown in figure 8.

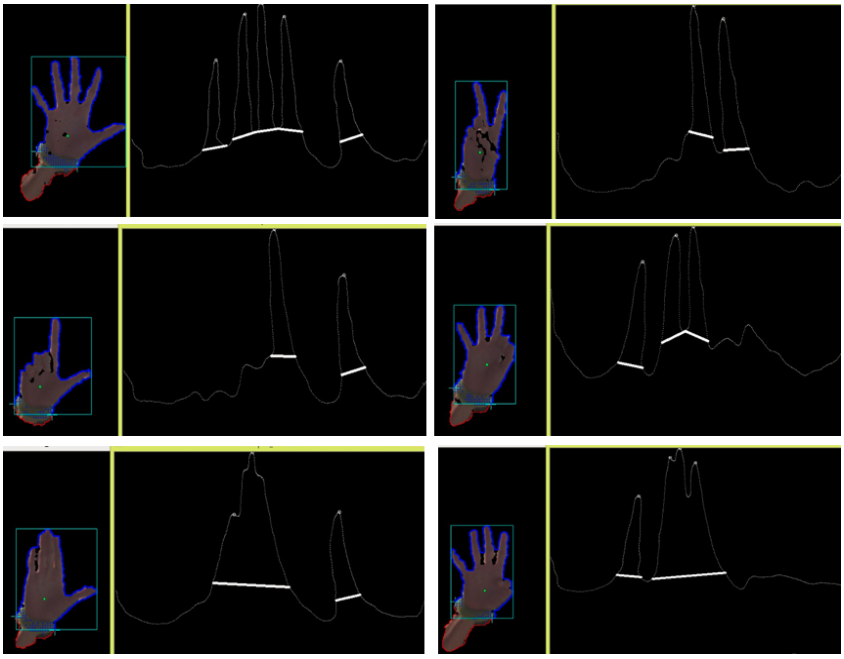

Fig. 8. Result of lasso algorithm

4. Conclusion

In this paper, we investigated how to extract hand gesture features from depth data in real-time via Kinect sensor. Firstly, we described the technical principles and features of the Kinect camera. And on this basis, we analysis the difficulties and key content on Kinect study. A real-time approach was proposed to extract the hand gesture feature by using image segmentation, Earth Mover's Distance and Lasso algorithms.

In the step of palm image segmentation, we use a contour length information based de-noise method, which is simple but practical. The hand gesture image has regular and stable contour, while the noise signal in the depth data is scattered and unstable, so the hand gesture feature always has longer contour than the noise signal. Therefore, significant effect can be achieved by setting a contour length threshold value. Then FEMD algorithm was proposed with new methods in the calculation of central point of palm image and the extraction of fingertip feature. Especially the proposed lasso algorithm can extract the fingertip feature from a hand contour curve correctly with excellent real-time performance. This work will provide a fundamental and distinctive feature for hand gesture analysis and recognition, especially in real-time human-computer interactions. Due to the feature extracted by this method is based on a single RGBD image, such finger features from continuous images can potentially form a dynamic and 3 dimensional feature, which could be very useful for dynamic gesture analysis and recognition. Future research will be focused on application in effective dynamic hand gestures and their recognition via the proposed hand features [19-21].

5. Appendix

Algorithm 1 Hand Edge Contour Generation

Require: Pixel[r] {Pixel is a 4 dimensional dataset with n pixels in the RGB-D image and r = {1,…,n}, n is the total number of the pixels in the image}
Require: Hand_centre {Hand_centre is the location of the centre of the tracked hand }
Require: initial_depth {initial_depth is the average depth of the tracked hand}
mini_depth = initial_depth-offset {offset is the user chosen threshold}
max_depth = initial_depth+offset
for all i such that 0≤i≤n **do**
  if mini_depth ≤depth(i) ≤ max_depth and |pos(i)-Hand_centre| ≤0.5*body_length {body_length is the preset length of the body in pixels; depth(i) returns the depths of pixcel[i]; pos() gets the position of pixel[i], } **then**
    is_hand[i] ← true { is_hand[] is an array that indicates hand pixels}
  else
    is_hand[i] ← false
  end if
end for
for all i such that 0≤i≤n **do**
  if Is_hand[i]==true **then**
    K ← {k1,…,k8| $|pos(k_r) - pos(i)| < 2$ } {get the eight nearby pixels around the ith pixel, 0≤r≤n}
    if $\prod_{j=1}^{8} Is\_hand[k_j] == true$ **then**
      Is_contour[i] ← true {Is_contour[] is an array that indicates the hand contour}
    else
      Is_contour[i] ← false
    end if
  else
    is_contour[i] ← false
  end if
end for
return is_contour

Algorithm 2 Lasso Algorithm

```
struct  m_signature {structure of fingertip feature}
{
    int a,b {the coordinate values of the fingers}
    bool R_suspended,L_suspended {indicates if the lasso is suspended}
    double rope_length  {the length of the lasso}
}

Dim m_signature_num As int {number of the fingertip signature}
Dim L_cur As int {left cursor of the lasso}
Dim R_cur As int {right cursor of the lasso}
Dim R_cur_move As bool
Dim L_cur_move As bool {if the cursor should be moved}
```

```
Dim distance As int {the dynamic length of the lasso}
Dim pre_distance As int {the former dynamic length of the lasso}
Rope(m_signature[],m_signature_num,L_cur,R_cur,R_cur_move,distance,pre_distance)
{function of the lasso algorithm}
{
For all i such that 0≤i≤m_signature_num {m_signature_num is the number of fingertip
features} then
    While distance<m_signature[i].rope_length {if the length of the lasso is longer than the
    preset value of the width of the finger,that means the lasso has clamped completely, the
    lasso procedure will come to an end}
    Do
       If  (R_suspended) AND (R_cur_move) then
            R_cur<-R_cur+1 {move R_cur right}
            calculate R_suspended
            If  R_suspended==ture {if the right side of the lasso is suspended, then its left side
            will move left} then
                R_cur_move ⇐ false
                calculate distance
            end if

            If (distance>pre_distance) AND (!L_suspended)  {if the length of the lasso
            become longer, then its left side will move left} then
                R_cur_move ⇐ false
                pre_distance ⇐ distance
            Else If (!L_suspended)AND(L_cur_move) then
                L_cur ⇐ L_cur+1
                calculate L_suspended
            end if

            If L_suspended==true {if the left side of the lasso is suspended, then move the
            right side towards the right next} then
                R_cur_move ⇐ true
                calculate distance
            end if
            If (distance>pre_distance&&!R_suspended)  {if the length of the lasso become
            longer,then move the right side towards the right} then
                R_cur_move ⇐ true
                pre_distance ⇐ distance
            end if
         end if
    end while

    m_signature[i].a ⇐ L_cur
    m_signature[i].b ⇐ R_cur
end for
```

6. Acknowledgement

The authors would like to acknowledge support from DREAM project of EU FP7-ICT 611391 and Research Project of State Key Laboratory of Mechanical System and Vibration China MSV201508.

7. References

[1] Khoshelham, Kourosh, and Sander Oude Elberink. "Accuracy and resolution of kinect depth data for indoor mapping applications." Sensors 12.2 (2012): 1437-1454.

[2] P. Dolla´r, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, IEEE, 2005, pp. 65–72.

[3] H.-M. Zhu, C.-M. Pun, Hand gesture recognition with motion tracking on spatial-temporal filtering, in: Proceedings of the 10th International Confer- ence on Virtual Reality Continuum and Its Applications in Industry, ACM, 2011, pp. 273–278.

[4] H. Farid, E. P. Simoncelli, Optimally rotation-equivariant directional deriva- tive kernels, in: Computer Analysis of Images and Patterns, Springer, 1997, pp. 207–214.

[5] Han, J., et al. "Enhanced computer vision with microsoft kinect sensor: a review." IEEE transactions on cybernetics 43(5): 1318-1334

[6] Ren, Zhou, Junsong Yuan, Jingjing Meng and Zhengyou Zhang,"Robust Part-Based Hand Gesture Recognition Using Kinect Sensor" IEEE Trans. On Multimedia, vol. 15, no. 5, August 2013.

[7] Ren, Zhou et al. "Minimum near-convex decomposition for robust shape representation." Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[8] Ju, Z., Wang, Y., Chen. S.Y., Liu, H. (2013) Depth and RGB Image Alignment for Hand Gesture Segmentation using Kinect, Proc. International Conference on Machine Learning and Cybernetics, pp. 1-8, Tianjing, China

[9] Alexiadis, Dimitrios S., et al. "Evaluating a dancer's performance using kinect-based skeleton tracking." Proceedings of the 19th ACM international conference on Multimedia. ACM, 2011.

[10] Matyunin, Sergey, et al. "Temporal filtering for depth maps generated by kinect depth camera." 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011. IEEE, 2011.

[11] Tara, R., P. Santosa, and T. Adji. "Hand segmentation from depth image using anthropometric approach in natural interface development." Int. J. Sci. Eng. Res 3.5 (2012): 1-4.

[12] Liang, Hui, Junsong Yuan, and Daniel Thalmann. "3D fingertip and palm tracking in depth image sequences." Proceedings of the 20th ACM international conference on Multimedia. ACM, 2012.

[13] Ren, Zhou, et al. "Robust hand gesture recognition with kinect sensor." Proceedings of the 19th ACM international conference on Multimedia. ACM, 2011.

[14] Keskin, Cem, et al. "Real time hand pose estimation using depth sensors." Consumer Depth Cameras for Computer Vision. Springer London, 2013. 119-137.

[15] C Li, H Ma, C Yang, M Fu, Teleoperation of a virtual iCub robot under framework of parallel system via hand gesture recognition, Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on, 1469-1474

[16] Tang, Matthew. "Recognizing hand gestures with microsoft's kinect." Palo Alto: Department of Electrical Engineering of Stanford University:[sn] (2011).

[17] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." Computer Vision–ECCV 2006. Springer Berlin Heidelberg, 2006. 404-417.

[18] B. Wang, C. Yang, and Q. Xie, Human-machine Interfaces based on Electromyography and Kinect applied to Teleoperation of a Mobile Humanoid Robot, the 10th World Congress on Intelligent Control and Automation (WCICA), pp. 3903-3908, Beijing, China, July 6-8, 2012

[19] Ju, Z. and Liu, H. Fuzzy Gaussian Mixture Models, Pattern Recognition, 45(3): 1146-1158, 2012;

[20] Ju, Z. and Liu, H. A Unified Fuzzy Framework for Human Hand Motion Recognition, IEEE Transactions on Fuzzy Systems, 19(5):901-913, 2011

[21] H. Reddivari, C. Yang*, Z. Ju, P. Liang, Z. Li and B. Xu, Teleoperation Control of Baxter Robot using Body Motion Tracking, the 2014 IEEE International Conference on Multisensor Fusion and Information Integration, Beijing, China, September 28-30, pp. 1-6, 2014