An Integrative Framework of Human Hand Gesture Segmentation for Human Robot Interaction

Zhaojie Ju, *Member, IEEE,* Xiaofei Ji, Jing Li, Honghai Liu, *Senior Member, IEEE,*

***Abstract*—This paper proposes a novel framework to segment hand gestures in RGB-D images captured by Kinect using human-like approaches for human-robot interaction. The goal is to reduce the error of Kinect sensing and consequently to improve the precision of hand gesture segmentation for robot NAO. The proposed framework consists of two main novel approaches. Firstly, the depth map and RGB image are aligned by using the genetic algorithm to estimate key points, and the alignment is robust to uncertainties of the extracted point numbers. Then a novel approach is proposed to refine the edge of the tracked hand gestures in RGB images by applying a modified Expectation-Maximisation (EM) algorithm based on Bayesian networks. The experimental results demonstrate the proposed alignment method is capable of precisely matching the depth maps with RGB images, and the EM algorithm further effectively adjusts the RGB edges of the segmented hand gestures. The proposed framework has been integrated and validated in a system of human-robot interaction to improve NAO robot's performance of understanding and interpretation.**

***Index Terms*—RGB-D, Alignment, Hand Gesture Segmentation, HCI**

I. Introduction

Recently, the problem of acquisition and recognition of human hand gestures from RGB-Depth (RGB-D) sensors, such as Microsoft's Kinect, is an important subject in the area of the computer vision and pattern analysis. In order to extract and recognise hand gestures from RGB-D data, many researchers conducted significant contribution, including the hand gesture extracting, tracking, recognising and so on [1]–[3]. These achievements are of much importance for research in areas of human-computer interaction (HCI). Researchers largely welcome the Kinect developed by Microsoft Corporation, as it can simultaneously acquire data of RGB image and depth map of the scene by its IR emitter and camera sensors. Its broad applications cover 3D reconstruction [4], [5], image processing [6], human-machine interface [2], [7]–[10], robotics [11], [12], object recognition [13], [14], just to name a few [15], [16]. However, there are many problems such as distortion and disaccord of depth and RGB images in corresponding pixels, especially the limitations in extracting of correct human hand gestures [17]. Due to the noises and holes in the RGB-D data, precisely segmenting the hand gestures is still a challenge.

The authors would like to acknowledge support from DREAM project of EU FP7-ICT (611391), Research Project of State Key Laboratory of Mechanical System and Vibration China (MSV201508), and NSFC (61463032).

Z. Ju and H. Liu are with School of Computing, University of Portsmouth, UK; X. Ji is with School of Automation, Shenyang Aerospace University, China; J. Li is with School of Information Engineering and Jiangxi Provincial Key Laboratory of Intelligent Information Systems, Nanchang University, China. Corresponding author: Honghai Liu, honghai.liu@port.ac.uk.

Manuscript received xxxx; revised xxxx.

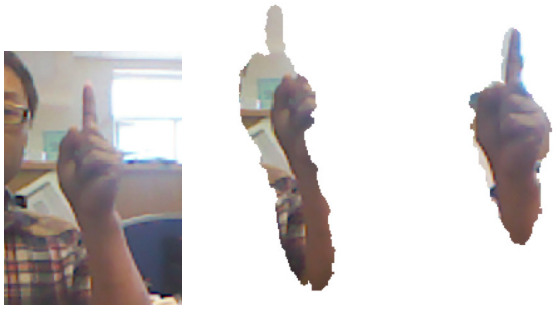
In computer vision, camera calibration is a necessary step in scene reconstruction in order to extract metric information from images [18]. This includes internal calibration of each camera as well as external parameters of relative pose calibration between the cameras. Colour camera calibration has been studied extensively and different calibration techniques have been developed for depth sensors depending on the circumstances [3], [19], [20]. In a similar manner, the calibration of RGB image and depth map is much essential for their consistency and synchronisation. For recovering and tracking the 3D position, orientation and full articulation of a human hand from markerless visual observations, an algorithm of minimising the discrepancy between the appearance and 3D structure of hypothesised instances of a hand model and actual hand observations was developed in [21]. Li implemented a novel algorithm for contactless hand gesture recognition, and it is a real-time system which detects the presence of gestures, to identify fingers and to recognise the meanings of nine gestures in a predefined popular gesture scenario [22]. For handling the noisy hand shapes obtained from the Kinect sensor, Zhang designed a approach of distance metric for hand dissimilarity measure, called Finger-Earth Mover's Distance [18]. As it only matches fingers while not the whole hand shape, it can better distinguish hand gestures of slight differences. In [23], Van *et al.* designed a robust and real-time system of 3D hand gesture interaction with a robot for understanding directions from humans. The system was implemented to detect hand gestures in any orientation and more in particular pointing gestures while extracting the 3D pointing direction.

Because of the complexity and dexterity of the human hand, recognising the unconstrained human hand motions is a fundamental challenge in existing algorithms [24]. Kinect provides a promising way to realise stable, effective and natural human-computer interaction [1], [25]–[27]. The rest of this paper is organised as follows. The problem of hand gesture segmentation via Kinect is given and the proposed framework is introduced in Section II; Depth and RGB image alignment is investigated in Section III. Hand gesture segmentation using an EM algorithm is proposed in section IV. Experimental results are discussed in Section V, followed by conclusions in Section VI.

II. Problems of Hand Gesture Segmentation via Kinect

Depth and colour/RGB images are simultaneously captured by Kinect at a frame rate of up to 30 fps. More than 300,000 depth-coloured points are captured in each frame. One "perfect" frame will consist of these points with absolutely correct

alignment of the depth and colour data. However, due to the limitations of the systematic design and the random errors, the alignment of the depth and RGB images highly relies on the identification of the mathematical model of the measurement and the calibration parameters involved. The characterisation of random errors is important and useful in further processing of the depth data, for example in weighting the point pairs or planes in the registration algorithm [28]–[30].



(a) RGB (b) Uncalibrated (c) Calibrated

Fig. 1: Hand gesture segmentation (a) RGB image; (b)uncalibrated; (c) Calibrated using official calibration

A proprietary algorithm is used to calibrate Kinect devices when manufacturing, and these calibrated parameters stored in the devices' internal memory are used to perform the image construction. The official calibration is adequate for human body motion analysis or casual use, but it lacks accuracy in hand gesture segmentation and recognition. Fig. 1b shows the result of hand segmentation based on depth threshold without official calibration from the RGB image in Fig. 1a, and it shows the colourful finger can not be seen and the mismatch between the depth and colour images is huge. Fig. 1c shows the result using the official calibration. It clearly indicates that only half of the finger can be seen in the segmented RGB image, and this will severely affect the further hand gesture recognition. Other calibration algorithms have been proposed to solve the problem of the disparity/depth distortion [20], *e.g.* Smisek *et al.* [31] introduced a depth distortion correction component as the average of the residuals in metric coordinates, while Daniel *et al.* [3] proposed a disparity distortion correction that depends on the observed disparity which further improves accuracy. These algorithms require a lot calibrating images and the optimisations are based on the whole scene, which means they are not practical and may sacrifice the precision of local space to achieve an overall minimisation. Since depth range of the Kinect devices is around 50cm to 5m and the resolution is about 1.5mm at 50cm and 5cm at 5m, the hand, as a small part of the body, needs to be closer to the camera to get a clearer image and it asks for higher precision in depth and RGB image alignment for hand segmentation and then for hand gesture recognition. In addition, due to the noise and holes of the depth data, the image segmentation based on the depth information has lots of mismatched pixels including the background pixels in the segmented objects and object pixels left in the background [16]. This problem with mismatched pixels has not been addressed in the current literature. Recently, more advanced methods have been reported to recognise hand gestures [32]. Fabio et al. introduced an effective way of exploiting depth information for hand gesture recognition, with a limited and not always required colour information aid for hand identification only, and achieved a very high recognition rate [33]. It used finger distance from the hand centroid as feature, which however is not always available as the fingertips might not be found due to occlusion or noise. Yuan et al. proposed a novel framework for recognising hand gestures, which is inspired by the current depth image-based body pose estimation technology, via a semiautomatic labeling strategy using a Kinect sensor and coloured glove. The accuracy of the recognition is limited by the hand segmentation and hand part classification [34].

The resolution of the kinect depth image is 640×480, which works well to track human body gestures. For smaller object, e.g. the highly articulated human hand which takes up only a small part in the whole image, it is very hard to detect and segment through the depth image. Based on the captured depth data from Kinect, e.g. in Fig. 2, there are a lot of noise with missing bits and flickering issue [35], [36]. These noises and holes will effect feature extraction and pattern recognition [37]. This paper focuses on precise segmentation of the hand

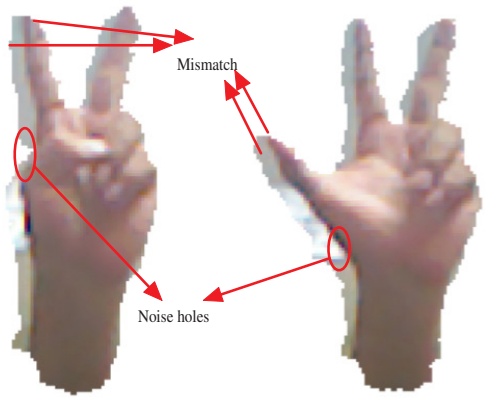


Fig. 2: The holes and mismatches in the hand images captured by Kinect

gestures using RGBD image, trying to get rid of the mismatch and holes. It will potentially provide help to extract hand features and further to strengthen recognition accuracy. In this paper, an integrative framework is proposed to precisely segment the hand gestures using RGBD image, shown in Fig. 3, similar to the human's approach, which normally tracks and locates the both hands based on the global human body gesture firstly and then extracts the details of the fingers based on the local colour clues. In this framework, genetic algorithm is firstly used to match the depth mag with RGB image, and then an EM algorithm is proposed to further adjust the segmentation edge based on the depth map, RGB image and locations of the pixels. The localisation of the human hand is realised by spatio-temporal filtering method [38], [39] based on the filtering global interest points on the dynamic images. After the whole hand is located, the edge of the hand will be

precisely detected and refined through assigning pixels around the edge with both the depth and RGB information. The contributions of this paper include two main approaches: 1) the proposed alignment method employs genetic algorithms to estimate the key points from both depth and RGB images, and it is robust to the uncertainties of the point numbers identified by using the common image processing tools such as corner and edge detectors. It is capable of correctly positioning the depth image with the RGB image. 2) due to the noise and holes in the depth map, the segmented result using the depth information has lots of mismatched pixels, which need further adjustment. To solve this problem, a novel approach has been proposed to further refine the edge of the segmented hand gesture via EM algorithm. The proposed approach has been further implemented to interact a humanoid robot using hand gestures.

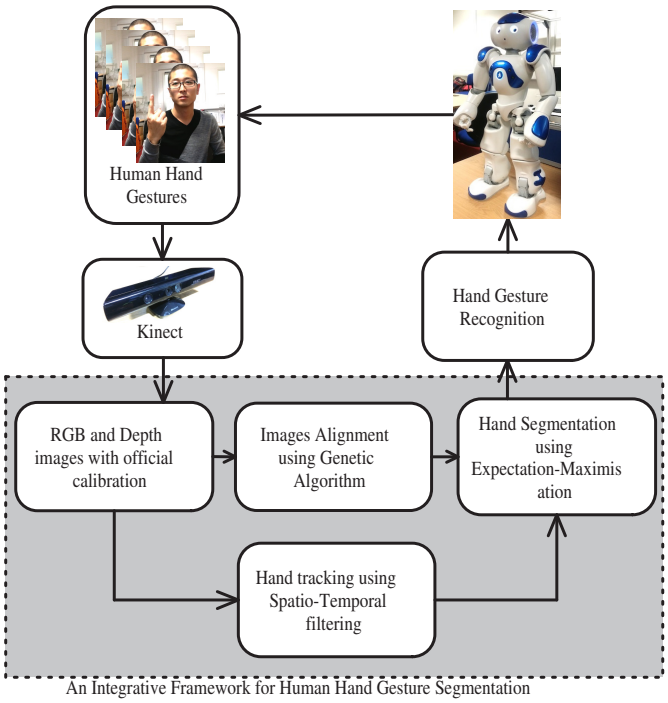


Fig. 3: An integrative framework of hand gesture segmentation with human-like approaches for human robot interaction.

III. Depth and RGB Image Alignment

A. Mathematical Model

Depth and RGB image alignment is essential for human motion analysis using Kinect, especially for the hand gesture recognition, which requires a much more accurate hand location and segmentation. It plays a key role in extracting motion features from the segmented images including both RGB and depth information.

Pinhole model is used to describe the RGB camera [3]. The calibration is to find the transformation matrix from 3D world coordinates to 2D image coordinates or between two 2D image coordinates by solving the unknown parameters of the camera model [40]. Let P be an arbitrary 3D point located in the scene and p_c be its projection on the RGB image plane. The coordinates of P in the RGB camera coordinate system are $[x_c, y_c, z_c]^T$ and in the world coordinate system are $[X, Y, Z]^T$. The coordinates of p_c in RGB image frame are $[u_c, v_c]^T$ and their relation can be expressed by the following transformation given by the homogeneous coordinates

$$\lambda_c \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} = F_c \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = A_c M_c \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

where λ_c is a scale factor and F_c is the perspective transformation matrix,

$$A_c = \begin{bmatrix} \alpha_c & \gamma_c & u_0^c & 0 \\ 0 & \beta_c & v_0^c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

A_c is the camera intrinsic parameter matrix; α_c and β_c are the scale factors in the RGB camera image coordinate system, (u_0^c, v_0^c) are the coordinates of the principal points, γ_c is the skewness of the two image axes and M_c is a 4 by 4 matrix describing the transformation from world coordinate system to camera coordinate system:

$$M_c = \begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \\ 0 & 1 \end{bmatrix} \tag{3}$$

where $\mathbf{t}_c = [t_x^c, t_y^c, t_z^c]^T$ describes the translation between the two systems, and $\mathbf{R}_c$ is a 3 by 3 orthonormal rotation matrix which can be defined by the three Euler angles along three axis respectively.

The depth camera typically outputs an image with depth values, denoted by $p_d = [u_d, v_d, z_d]^T$, where (u_d, v_d) are the pixel coordinates, and z_d is the depth value. The mapping from p_d to the point in the world coordinate system $[X, Y, Z]^T$ follows a similar model to that used for the RGB camera:

$$\lambda_d \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = A_d M_d \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4}$$

where A_d is the depth camera's intrinsic parameter matrix. If the same point P is captured simultaneously by the RGB and depth cameras, according to the Eq. 1 and 4, the transformation between the coordinates in the RGB and depth camera coordinate systems can be expressed as

$$\begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} = K_c H_{dc} K_d^{-1} \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} \tag{5}$$

where the homographs matrix H_{dc} is

$$H_{dc} = R - \frac{t n^T}{d} \tag{6}$$

R is the rotation matrix by which the depth camera is rotated in relation to the RGB camera; t is the translation vector from the depth camera to the RGB camera; n and d are the normal vector of the plane and the distance to the plan respectively. $K_c = A_c \begin{bmatrix} I \\ 0 \end{bmatrix}$; $K_d = A_d \begin{bmatrix} I \\ 0 \end{bmatrix}$ are the cameras' intrinsic parameter matrices.

Based on the above calibration model, we can use key points P_i in the scene to estimate the transformation matrix, $T_{dc} = K_c H_{dc} K_d^{-1}$, which translate depth image coordinates to RGB image coordinates. Key points identified in both RGB and depth map are used to build the mapping relationship between them. Suppose there are m key points $p^i = [u^i, v^i, 1]^T, 1 \le i \le m$, captured simultaneously by both cameras, one estimation of the transformation matrix can be achieved by considering any three points and the overall estimation can be found by

$$\bar{T_{dc}} = \frac{\sum_{1 \le i \ne j \ne k \le m} \begin{bmatrix} u_c^i & u_c^j & u_c^k \\ v_c^i & v_c^j & v_c^k \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} u_d^i & u_d^j & u_d^k \\ v_d^i & v_d^j & v_d^k \\ 1 & 1 & 1 \end{bmatrix}}{\binom{3}{m}} \tag{7}$$

(a) RGB image (b) Depth image

Fig. 4: Checkerboard captured via Kinect

A checkerboard is employed and displaced around 50cm to 100cm in front of the Kinect, and the distance of the checkerboard needs to be adjusted properly to achieve a satisfying performance of the crossing point and apex identification. It would be ideal to put the checkerboard about 0.80 meters away from the Kinect as it is the place where the Kinect can better track the hand with a proper resolution, shown in Fig. 4. The board consists of exact 5×4 black and white square boxes. Different from RGB images, depth images can not identify the crossing points automatically and thus these crossing points can not be regarded as the key points [20]. The four apexes of the checkerboard are selected as the key points in this paper. From the depth image of the checkerboard, we can easily achieve the edges of the board, based on which the four apexes can be estimated. In the RGB image, the crossing points can be automatically identified and the apexes can also be estimated. We employ genetic algorithm to estimate these key points.

B. Apex Estimation in the Depth Image

Fig. 5 shows the edges extracted by using the Sobel approximation to the derivative. It returns edges on those points where the gradient is maximum. Assume there are n extracted edge points $e_i = [e_i^x, e_i^y]^T$ where $1 \le i \le n$; the four estimated apexes are $a_i = [u_d^i, v_d^i]^T$, where $1 \le i \le 4$, and the four estimated edges are $E_1 = (a_1, a_2); E_2 = (a_2, a_3); E_3 = (a_3, a_4); E_4 = (a_4, a_1)$. The distance between ith extracted edge point e_i and all the estimated edges $\{E_i, i \in (1,2,...,4)\}$ is defined as the distance between this edge point with its nearest estimated edge:

$$D_i = min(d_i^1, d_i^2, d_i^3, d_i^4) \tag{8}$$

where $d_i^j = \frac{\left|(a_{j+1} - a_j) \times (a_j - e_i)\right|}{\left|a_{j+1} - a_j\right|}$ and $a_5 = a_1$.

(a) Depth image (b) Estimated edge points

Fig. 5: Checkerboard edge

To use the genetic algorithm to find the four apexes in depth image, the fitness function is set as follows:

$$\bar{D} = (\textstyle\sum_{i=1}^{n} D_i)/n \tag{9}$$

where $\bar{D}$ is the average distance between the extracted edge points and the estimated edges. To make the genetic algorithm find the solution more efficiently, the bounds for the four points are set as follows:

$$\begin{array}{l} e_x^{min} - 10 < u_d^1, u_d^4 < e_x^{mean}; \\ e_x^{mean} < u_d^2, u_d^3 < e_x^{max} + 10; \\ e_y^{min} - 10 < v_d^1, v_d^2 < e_y^{mean}; \\ e_y^{mean} < v_d^3, v_d^4 < e_y^{max} + 10; \end{array} \tag{10}$$

where $e_x^{min} = min(\{e_i^x, i = 1,2,...,n\})$ is the minimum of the x-coordinates in the extracted edge points; $e_x^{max} = max(\{e_i^x, i = 1,2,...,n\})$; $e_y^{min} = min(\{e_i^y, i = 1,2,...,n\})$; $e_y^{max} = max(\{e_i^y, i = 1,2,...,n\})$; $e_x^{mean} = \sum_{i=1}^{n} e_i^x/n$; $e_y^{mean} = \sum_{i=1}^{n} e_i^y/n$.

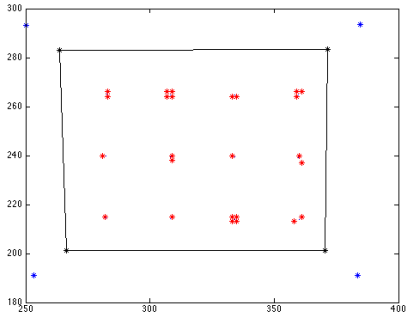
C. Apex Estimation in the RGB Image

To estimate the apexes in RGB image, first we need to extract the corner points. Fig. 6 shows the corners extracted by using the Harris corner detector. The estimated apexes based on the depth edge points are shown in blue stars in Fig. 6b from Sec. III-B. The corners in the middle the checkerboard can be easily selected using a sub-rectangle shown in black in Fig. 6b, which is achieved by shrinking the rectangle shaped by the blue stars. If there are g corners identified $c_i = [c_i^x, c_i^y]^T$ where $1 \le i \le g$; the four estimated apexes are $a_i = [u_d^i, v_d^i]^T$, where $1 \le i \le 4$. Thus the 12 estimated crossing point shown in Fig. 4a can be calculated from the four estimated apexes as:

$$r_{ij} = \frac{(4-i)\left[(5-j)a_1 + ja_2\right] + i\left[ja_3 + (5-j)a_4\right]}{20} \tag{11}$$

where $i \in (1,2,...4)$ and $j \in (1,2,3)$.

(a) Corners identified in RGB image



(b) Corners selected by the estimated apexes in Sec. III-B

Fig. 6: Checkerboard corners identification and selection

The distance between ith extracted corner c_i and all the estimated crossing points is defined as the distance between this corner with its nearest estimated crossing point:

$$D_i = min(\{d_i^j, \text{where j=1,2,...,12}\}) \tag{12}$$

where $d_i^j = norm2(c_i, a_j)$.

The fitness function for the genetic algorithm is set as follows:

$$\bar{D} = (\sum_{i=1}^{g} D_i)/g \tag{13}$$

where $\bar{D}$ is the average distance between the corners and the estimated crossing points. The bounds for the four points are set similarly as above, and the initial values of the four apexes are set as the estimated depth apexes achieved in the previous subsection, since they are supposed to be very close when employing official calibration of the Kinect. The distance of the checkerboard needs to be adjusted properly to achieve a satisfying performance of the crossing point and apex identification. It would be ideal to put the checkerboard about 0.80 meters away from the Kinect as it is the place where the Kinect can better track the hand with a proper resolution and at the same time the hand can move freely to perform motions and gestures.

IV. Hand Gesture Segmentation using EM Algorithm

The depth and RGB images have been roughly adjusted and aligned using the alignment method in Sec. III. However, due to the noises and holes in the RGB-D data, the colour map of the human hand can not be effectively segmented using only the depth information [16]. In this session, we will apply EM algorithm to further estimate the boundary of the hand gestures and more precisely segment hand images.

A. The proposed EM Algorithm

Each pixel in the Kinect image has RGB values, a depth value and its 2D location, based on which the estimation of the probability of this pixel belonging to the hand gesture can be expressed by $p(H=1|R,D,L)$ or $p(H|R,D,L)$. R is the pixel RGB value; D is the pixel depth value; L is the location of the pixel. H is a binary variable indicating whether a pixel belongs to a hand or not, when $H=1$ or H means this pixel belongs to a hand and $H=0$ or $\tilde{H}$ means this pixel does not. The events of R, D and L can be reasonably assumed to be independent, and according to the Bayesian Network, we can have

$$\begin{aligned} p(H|R,D,L) &= \frac{p(H)p(R|H)p(L|H)p(D|H)}{\sum_H p(H)p(R|H)p(L|H)p(D|H))} \\ &= \frac{p(H)p(R|H)p(L|H)p(D|H)}{p(H)p(R|H)p(L|H)p(D|H)+p(\tilde{H})p(R|\tilde{H})p(L|\tilde{H})p(D|\tilde{H})} \end{aligned} \tag{14}$$

where $p(H)$ is prior probability of the hand gesture; $p(R|H)$ is the probability of the RGB value given that this pixel is part of a hand and it assumes to be a Gaussian distribution with a mean of μ_{RH} and a covariance of Σ_{RH}; $p(D|H)$ is the probability of the depth value given this pixel is part of a hand and it assumes to be Gaussian distributed with a mean of μ_D and a covariance of Σ_D; $p(L|H)$ is the probability of the pixel location given this pixel belongs to a hand and its distribution is given below:

$$p(L|H) = \tfrac{1}{2}\left(erf(\tfrac{\text{dist(L)}}{\sqrt{2}\delta_L}) + 1\right) \tag{15}$$

where function erf is a Gauss error function:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2} \mathrm{d}t \tag{16}$$

and the function dist(L) is to get the minimum distance between the pixel and the hand edge. dist(L) is negative when the pixel is inside of the edge and positive when outside of the edge. Gauss error function is frequently used since it is obtained by integrating the normalised Gaussian distribution, which is often used in the natural and social sciences to represent real-valued random variables whose distributions are not known. $p(\tilde{H})$ is the probability of this pixel not belonging to a hand, and $p(\tilde{H}) = 1 - p(H)$; $p(R|\tilde{H})$ is the probability of the RGB value given that this pixel is part of the background and it assumes to be a Gaussian distribution with a mean of μ_{RB} and a covariance of Σ_{RB}; $p(D|\tilde{H})$ is the probability of the depth value given this pixel is part of the background and it assumes to be of uniform distribution $unif(depth_{min}, depth_{max})$, where the $depth_{min}$ is the minimum of the depth value in the scene and $depth_{max}$ is the maximum; $p(Location|\tilde{H})$ is the probability of the pixel location given this pixel belongs to the background and $p(L|\tilde{H}) = 1 - p(L|H)$.

The parameters are $\Theta = (\mu_{RH}, \Sigma_{RH}, \mu_{DH}, \Sigma_{D\tilde{H}}, \Sigma_{R\tilde{H}}, \mu_{D\tilde{H}})$. The resulting density for the samples is

$$p(\mathscr{U}|\Theta) = \prod_{t=1}^{n} p(x_t|\Theta) = \mathscr{L}(\Theta|\mathscr{U}) \tag{17}$$

where $\mathscr{U}$ means all the captured pixel information including the *RGB*, *Depth* and *Location* and $\mathscr{U} = \{u_1, \ldots, u_n\}$, $u_i = \{R_i, D_i, L_i\}$ and n is the number of the pixels.The function $\mathscr{L}(\Theta|\mathscr{U})$ is called the likelihood of the parameters given the data, or the likelihood function. The likelihood is considered as a function of the parameters Θ where the data $\mathscr{U}$ is fixed. In the maximum likelihood problem, the objective is to estimate the parameters set Θ that maximizes $\mathscr{L}$. That is to find Θ^* where

$$\Theta^* = \arg\max_{\Theta} \mathscr{L}(\Theta|\mathscr{U}) \tag{18}$$

Usually, the EM algorithm (e.g., [41], [42]) is proposed to maximise the $\mathscr{L}$. The iteration of an EM algorithm estimating the new parameters in terms of the old parameters is proposed and given as follows:

- E-step: compute "expected" classes of all pixels for hand gesture and background, $p(H|R_t, D_t, L_t)$ and $p(\tilde{H}|R_t, D_t, L_t)$ using Eq. 14.
- M-step: compute maximum likelihood given the pixel class membership distributions according to equations 19-24.

$$p(H)^{new} = \frac{1}{n} \sum_{t=1}^{n} p(H|R_t, D_t, L_t); \tag{19}$$

$$p(\tilde{H})^{new} = 1 - p(H) \tag{20}$$

$$[\mu_{RH}^{new}, \mu_{DH}^{new}] = \frac{\sum_{t=1}^{n} p(H|R_t, D_t, L_t)[R_i, D_i]}{\sum_{t=1}^{n} p(H|R_t, D_t, L_t)} \tag{21}$$

$$\Sigma_{RGBH}^{new} = \frac{\sum_{t=1}^{n} p(H|R_t, D_t, L_t)(RGB_t - \mu_{RH}^{new})(R_t - \mu_{RH}^{new})^T}{\sum_{t=1}^{n} p(H|R_t, D_t, L_t)} \tag{22}$$

$$\Sigma_{DH}^{new} = \frac{\sum_{t=1}^{n} p(H|R_t, D_t, L_t)(D_t - \mu_{DH}^{new})(D_t - \mu_{DH}^{new})^T}{\sum_{t=1}^{n} p(H|R_t, D_t, L_t)} \tag{23}$$

$$edge^{new} = f(p(H|R, D, L)) \tag{24}$$

where μ_{RH} and Σ_{RH} are the new estimated mean and covariance of the hand in RGB values; μ_{DH} and Σ_{DH} are the new estimated mean and covariance of the hand in depth values; $f()$ is the function to estimate the new edge of the hand according to the probabilities of all pixels belonging to the hand gesture, and its details are given in Sec. IV-B .

B. Edge Estimation

The probability of $p(H|R_i, D_i, L_i)$ can be normalised as:

$$p'(H|R_i, D_i, L_i) = \begin{cases} 0, & if\ p(H|R_i, D_i, L_i) < 0.01 \\ 1, & if\ p(H|R_i, D_i, L_i) > 0.99 \\ p(H|R_i, D_i, L_i), & else \end{cases} \tag{25}$$

according to $p'(H|R_i, D_i, L_i)$, we can easily have two edges: external edges $\{x_i^E\}, i = 1, \ldots, n^E$ for all pixels whose probabilities are less than 0.01 and internal edges $\{x_i^I\}, i = 1, \ldots, n^I$ whose probabilities are more than 0.99. n^E and n^I are the number of the edge points on external edge and internal edge respectively. A proper threshold needs to be chosen to balance the effectiveness and efficiency of the EM algorithm. Based on experimental results, a threshold of 0.01 is chosen to assign "definite" hand and "definite" background pixels. For each point x_i^E on the external edge, there is a point x_j^I who has a minimum distance between x_i^E and the internal edge points; similarly, for each point x_i^I on the internal edge, there is a point x_j^E who has a minimum distance between x_i^I and the internal edge points. Assume the points pairs $(x_i^I, x_i^E), i = 1, \ldots, n^P$ are the unique pairs with such minimum distances, n^P is the number of those unique pairs. Assume a line l_i is determined by the pair points (x_i^I, x_i^E), we can find the pixels x_k^i which are near to this line and whose probabilities are less than 0.99 and more than 0.01, as shown in green circle in Fig. 7. Based on the near points and their projection point, we can estimate the edge model on this line by

$$[\delta_i, a_i] = \arg\min_{\delta_i, a_i} \sum_{k=1}^{n^P} \left(\frac{1}{2} \left(\mathrm{erf}\left(\frac{D(pr(x_k^i), x_i^E) - a_i}{\sqrt{2}\delta_i} \right) + 1 \right) - p(H|x_k^i) \right) \tag{26}$$

where $pr(x_k^i)$ is the projection point of x_k^i and $D(x_i, x_j)$ is the distance between the location x_i and x_j; The minimum problem can be solved by Least-Square Fitting method. One example of the fitting results is given in Fig. 8. Then the edge point on the line l_i can be found by

$$edge_i^{new} = \frac{a_i}{D(x_i^I, x_i^E)}(x_i^I - x_i^E) + x_i^E \tag{27}$$

as shown in red circle in Fig. 7.

C. Implementation

To initialise the parameter set Θ, the hand gesture will be segmented based only on the depth information. Firstly, Spatial-temporal filtering (STF) [38], [39] 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. 9. The initial edge of the hand gesture can be achieved using Sobel method [43]. The pixels in the hand edge belong to hand with a full probability to the hand gesture, $p(H|R_i, D_i, L_i) = 1$, and others have a full probability to the background, $p(\tilde{H}|R_i, D_i, L_i) = 1$. The parameter set can be achieved by equations 19 to 23. The EM algorithm for segmenting the hand gesture is shown in Algorithm 1, where the threshold is set to stop the iteration of the EM algorithm with an acceptable error. The smaller the threshold is, more precise the fitting of the EM algorithm will be and more computational cost will be taken.

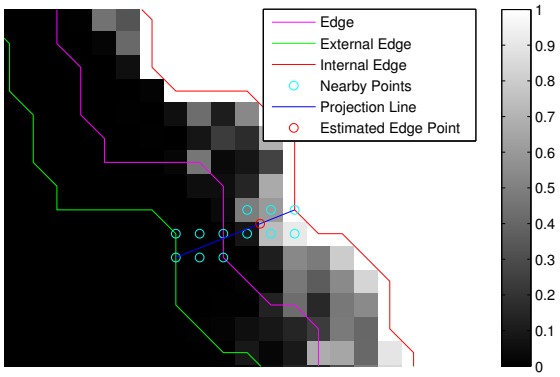


Fig. 7: An example shows three edge lines (external edge in green, original edge in magenta and internal edge in red) based on the probabilities of the pixels belonging to a hand (the probability is shown in a grayscale), and the estimated edge point in red circle has been identified based on the pixels in green circles near to the projection line

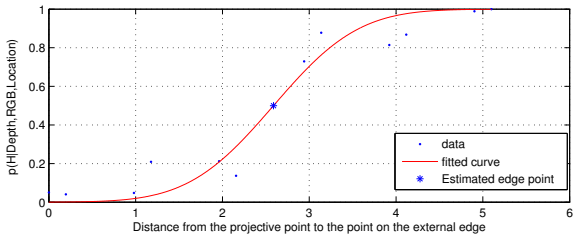


Fig. 8: An example of the fitting result. The projection points on the line are in blue dots; the fitting curve is in red line; the estimated edge point is in blue star.

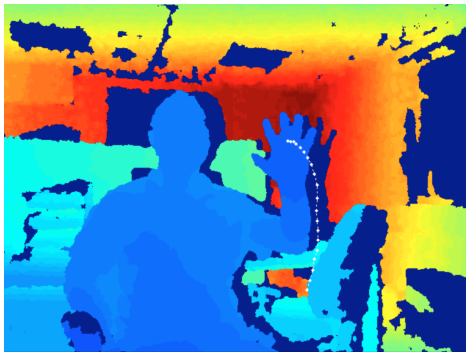


Fig. 9: Hand tracking using Spatio-Temporal filtering [38], [39], and the hand trajectory is shown in white dots.

V. Experiment Results

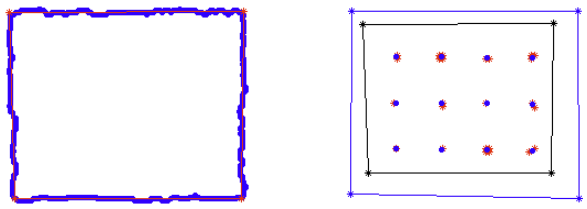
A. Alignment

The above algorithm has been implemented in Matlab. Various data have been collected and vaulted to show its performance. Genetic algorithm can always find the best

Algorithm 1 EM algorithm to segment hand gesture

Require: Fix R {R is the depth range used to segment hand gesture via only depth information.}.

1: $D^0 \leftarrow STF$ [38], [39] {Use Spatial-temporal filtering to track the hand and get the hand depth value}
2: $p(\tilde{H}|R_i, D_i, L_i) = 1/0$ {Use Sobel method to get the edge of the hand according to the R and D^0, and set the initial probability for each pixel}
3: **repeat**
4: $\{p(H)^{new}, p(\tilde{H})^{new}\} \leftarrow Eq.19$ and 20 {Compute the new prior probabilities of the hand gesture and background using Eq. 19 and 20}
5: $\{\mu_{RJ}^{new}, \mu_{RJ}^{new}\} \leftarrow Eq.$ 21{Compute the new RGB and Depth centres of the hand gesture and background using Eq. 21}
6: $\{\Sigma_{RJ}^{new}, Sigma_{DH}^{new}\} \leftarrow Eq.$ 22 and 23 {Compute the RGB variance of the hand gesture and background using Eq. 22 and the Depth variance of the hand gesture using Eq. 23}
7: $edge^{new} \leftarrow Eq.$ 24{Get the new edge using Eq. 24}
8: $p^{new}(H|R, D, L) \leftarrow Eq.$ 14{Upgrade the probabilities using Eq. 14 }
9: $\log(\mathscr{L}(\Theta|\mathscr{U})^{new}) \leftarrow Eq.$ 17{Compute the log-likelihood using Eq. 17}
10: **until** $\frac{\log(\mathscr{L}(\Theta|\mathscr{U})^{new}}{\log(\mathscr{L}(\Theta|\mathscr{U})^{old}} - 1 \leq threshold$ {Stop if the relative difference of the log-likelihood between two adjacent iterations is below the preset threshold}



(a) Four apexes (red stars) found in the depth image

(b) Four apexes (blue stars) found in the RGB image

Fig. 10: Solutions of the genetic algorithm

solution due to the pre-set searching bound for each variable and the close precise initialisation. One example of the genetic algorithm results for the above depth images is shown in Tab. I. The best average distance, 0.767 pixel, is found after five generations. The four apexes for both depth and RGB images are shown in Fig. 10a and 10b respectively. It demonstrated that the proposed algorithm is able to find the best key points based on the images captured. In addition, the numbers of edge points are not constant and the corners identified in the RGB image are more than 12 crossing points on the checkerboard, which may cause problems for the algorithms using the edge points/corners as the key points. The method in this paper uses four estimated apexes instead of the edge points/corners as the key points, and can find the optimised solution independent of the numbers of edge points or corners.

Fig. 11a compares these estimated apexes in the RGB

(a) “One” (b) “Two” (c) “Three” (d) “I love you” (e) “Good luck”

Fig. 13: Segmentation results of hand gestures “one”, “two”, “three”, “I live you” and “good luck” using different methods. Row one: segmented gestures with official calibration; row two: segmented gestures with the proposed alignment method; row three: segmented gestures with the proposed alignment and EM algorithm, and their iterations are 4, 6, 4, 5 and 4 respectively with a threshold of 1e-4.

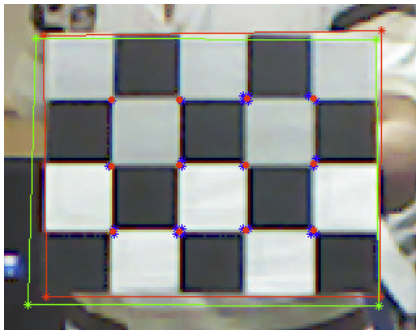
Generation	f-count	Best f(x)	Max Constraints	Stall Generations
1	1060	1.06589	0	0
2	2100	0.793731	0	0
3	3140	0.784879	0	0
4	4180	0.767132	0	0
5	5220	0.76685	0	0

TABLE I: Depth apexes optimisation using genetic algorithm

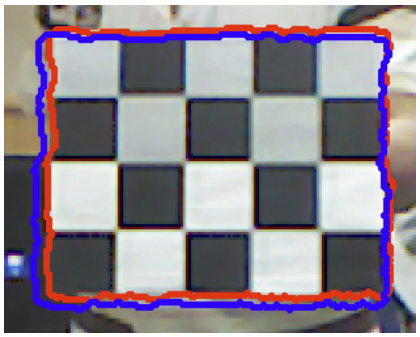
image. The difference between them will be used to determine the transformation matrix from the depth coordinates to RGB coordinates. The final transformation is achieved through Eq. 7. Then we transform the depth edge into RGB image coordinate system shown in red in Fig. 11b and the original depth edge is in blue. It is clear to see that using the red edge to segment the checkerboard is much better than the blue.

B. Hand Gesture Segmentation

Hand gesture segmentation has been evaluated based on the proposed alignment method. Improvements have been achieved and segmentation results of hand gesture “five” are shown with the comparison between the official calibration and the proposed alignment algorithm in Fig. 12a and 12b, where the proposed algorithm corrects the alignment of the depth image with RGB image, and almost all the coloured fingers have been extracted. It can also see that though the segmented hand gesture shown in Fig. 12b is much better aligned than the one in Fig. 12a, there are still some mismatched pixels, some of which belonging to the background are selected as



(a) Comparison between Apexes estimated in depth image (green) and RGB image (red)



(b) Comparison between transformed depth edge points (red) and original edge points (blue)

Fig. 11: Data comparison

(a) With the official calibration method

(b) with the proposed alignment method

(c) with the EM algorithm with 2 iterations

(d) with the EM algorithm with 4 iterations

Fig. 12: Segmentation results of the gesture "five" using different methods

hand pixels and some of which being part of the hand are misplaced into the background. To correct these mismatched pixels, the edge of the segmented hand gesture is further refined by the proposed EM algorithm and the results for the gesture "five" are given in Fig. 12c, which shows the result of the EM algorithm with 2 iterations. Fig. 12d shows the result of the EM algorithm with 4 iterations. The refined hand gestures contains less mismatched points and are much cleaner than those in figures 12a and 12b. Results on five other hand gestures, (*i.e.* "one", "two", "three", "I love you" and "good luck") are shown in Fig. 13, where gestures in the first raw are the segmented hand gestures with official calibration, the ones in the second raw are results with only the proposed alignment method, and the third raw gives the final refined results by further applying the proposed EM algorithm on the aligned gestures from the second raw.

C. Implementation in Human Robot Interaction

NAO is the most widely used humanoid robot for academic purposes worldwide, which is fully interactive, fun, and constantly evolving [44]. NAO also offers the flexibility for developing and attracting more interdisciplinary research projects in the near future. Many sensors and actuators on NAO, convenient size, and attractive appearance, combined with sophisticated embedded software, make it a unique humanoid robot ideal for many research fields. However, cameras on the NAO are not suitable to recognise human hand gestures due to low resolution and limited computing speed [45]. As a mature commercial produce, Kinect has been used extensively to understand and recognise human motions. In this work, Kinect has been integrated with NAO robot to strengthen its capabilities to understand and interpret human hand gestures, with the help of proposed framework, shown in Fig.14. Not only can NAO respond to human voice commends, but it is also able to react effectively to human hand gestures.

Fig. 14: The proposed integrative framework is implemented and evaluated to interact with a humanoid robot, NAO.

VI. Concluding Remarks

In this paper, a novel integrative framework has been proposed to segment hand gestures in RGB-D data using the Kinect device. Image alignment and refinement have been addressed in this framework to improve the precision of hand segmentation based on human-like approaches. The proposed alignment method employs the genetic algorithm to estimate the key points from both depth maps and RGB images, and it is robust to the uncertainties of the point numbers identified by using the common image processing tools such as corner and edge detectors. It is capable of correctly positioning the depth image with the RGB image. However, due to the noise and holes in the depth map, the segmented result using only the depth information has lots of mismatched pixels, which need further adjustment. To solve this problem, a novel approach has been proposed to further refine the edge of the segmented hand gesture using a modified EM algorithm. The experimental results show that the results by the proposed methods precisely segment the hand gestures and are much better than the official calibrated images and the results with only the proposed alignment method. The proposed framework has been implemented and validated in a system combining the Kinect with the NAO robot. It provides a significant improvement to the performance of the hand segmentation, which will potentially contribute to hand gesture recognition in the human-robot interaction.

A quantitative validation will be further investigated to demonstrate the effectiveness of the proposed methods. Since the proposed EM algorithm is based on the pixels, which takes more time than traditional segmentation methods. The computational cost could be alleviated by the sampling strategies or faster convergence process introduced in [24], [42], [46]. Our future research will be on the efficiency improvement of the proposed methods to adapt to different environments and conditions in the real-time applications, such as the dynamic gesture interaction with humanoid robots.

References

[1] K. Khoshelham, S. O. Elberink, Accuracy and resolution of kinect depth data for indoor mapping applications, Sensors 12 (2) (2012) 1437–1454.

[2] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, R. Moore, Real-time human pose recognition in parts from single depth images, Communications of the ACM 56 (1) (2013) 116–124.

[3] C. Herrera, J. Kannala, et al., Joint depth and color camera calibration with distortion correction, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (10) (2012) 2058–2064.

[4] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, A. Fitzgibbon, Kinectfusion: Real-time dense surface mapping and tracking, in: IEEE International Symposium on Mixed and Augmented Reality, IEEE, 2011, pp. 127–136.

[5] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison, A. Fitzgibbon, Kinectfusion: real-time dynamic 3d surface reconstruction and interaction, in: ACM SIGGRAPH 2011, ACM, 2011, p. 23.

[6] N. Silberman, R. Fergus, Indoor scene segmentation using a structured light sensor, in: IEEE International Conference on Computer Vision Workshops, IEEE, 2011, pp. 601–608.

[7] J. Preis, M. Kessel, M. Werner, C. Linnhoff-Popien, Gait recognition with kinect, in: 1st International Workshop on Kinect in Pervasive Computing, 2012, pp. 1–6.

[8] J. L. Raheja, A. Chaudhary, K. Singal, Tracking of fingertips and centers of palm using kinect, in: Third International Conference on Computational Intelligence Modelling and Simulation, IEEE, 2011, pp. 248–252.

[9] W. Xu, E. J. Lee, Gesture recognition based on 2d and 3d feature by using kinect device, in: International Conference on Information and Security Assurance, Vol. 6, 2012.

[10] L. Shao, L. Ji, Y. Liu, J. Zhang, Human action segmentation and recognition via motion and shape analysis, Pattern Recognition Letters 33 (4) (2012) 438–445.

[11] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers, R. Siegwart, Towards a benchmark for rgb-d slam evaluation, in: Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conference, Vol. 2, 2011, p. 3.

[12] C. Li, H. Ma, C. Yang, M. Fu, Teleoperation of a virtual icub robot under framework of parallel system via hand gesture recognition, in: IEEE International Conference on Fuzzy Systems, IEEE, 2014, pp. 1469–1474.

[13] L. Bo, X. Ren, D. Fox, Unsupervised feature learning for rgb-d based object recognition, Experimental Robotics (2013) 387–402.

[14] L. Spinello, K. O. Arras, People detection in rgb-d data, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 3838–3843.

[15] Z. Zhang, Microsoft kinect sensor and its effect, Multimedia, IEEE 19 (2) (2012) 4–10.

[16] L. Cruz, D. Lucio, L. Velho, Kinect and rgbd images: Challenges and applications, in: SIBGRAPI Conference on Graphics, Patterns and Images Tutorials, IEEE, 2012, pp. 36–49.

[17] J. Han, L. Shao, D. Xu, J. Shotton, Enhanced computer vision with microsoft kinect sensor: A review, IEEE Transactions on Cybernetics 43 (5) (2013) 1318–1334.

[18] Z. Zhang, Flexible camera calibration by viewing a plane from unknown orientations, in: IEEE International Conference on Computer Vision, Vol. 1, IEEE, 1999, pp. 666–673.

[19] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, C. Theobalt, Performance capture of interacting characters with handheld kinects, in: European conference on Computer Vision-Volume Part II, Springer-Verlag, 2012, pp. 828–841.

[20] K. Berger, K. Ruhl, Y. Schroeder, C. Bruemmer, A. Scholz, M. A. Magnor, Markerless motion capture using multiple color-depth sensors., in: Vision, Modeling, and Visualization, 2011, pp. 317–324.

[21] I. Oikonomidis, N. Kyriazis, A. Argyros, Efficient model-based 3d tracking of hand articulations using kinect, in: British Machine Vision Conference, 2011, pp. 101–1.

[22] Y. Li, Hand gesture recognition using kinect, in: IEEE International Conference on Software Engineering and Service Science, IEEE, 2012, pp. 196–199.

[23] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, M. Buss, Real-time 3d hand gesture interaction with a robot for understanding directions from humans, in: IEEE International Symposium on Robot and Human Interactive Communication, IEEE, 2011, pp. 357–362.

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

[25] T. Gill, J. Keller, D. Anderson, R. Luke, A system for change detection and human recognition in voxel space using the microsoft kinect sensor, in: IEEE Applied Imagery Pattern Recognition Workshop, IEEE, 2011, pp. 1–8.

[26] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, P. Presti, American sign language recognition with the kinect, in: International conference on multimodal interfaces, ACM, 2011, pp. 279–286.

[27] M. Tang, Recognizing hand gestures with microsofts kinect, Palo Alto: Department of Electrical Engineering of Stanford University.

[28] S. Rusinkiewicz, M. Levoy, Efficient variants of the icp algorithm, in: Third International Conference on 3-D Digital Imaging and Modeling, IEEE, 2001, pp. 145–152.

[29] K. Khoshelham, Automated localization of a laser scanner in indoor environments using planar objects, in: International Conference on Indoor Positioning and Indoor Navigation, IEEE, 2010, pp. 1–7.

[30] K. Khoshelham, Accuracy analysis of kinect depth data, in: ISPRS workshop laser scanning, Vol. 38, p. W12.

[31] J. Smisek, M. Jancosek, T. Pajdla, 3d with kinect, in: Consumer Depth Cameras for Computer Vision, Springer, 2013, pp. 3–25.

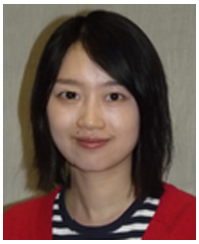
[32] S. S. Rautaray, A. Agrawal, Vision based hand gesture recognition for human computer interaction: a survey, Artificial Intelligence Review 43 (1) (2015) 1–54.
[33] F. Dominio, M. Donadeo, P. Zanuttigh, Combining multiple depth-based descriptors for hand gesture recognition, Pattern Recognition Letters 50 (2014) 101–111.
[34] Y. Yao, Y. Fu, Contour model based hand-gesture recognition using kinect sensor, IEEE Transactions on Circuits and Systems for Video Technology 99 (2014) 1–10.
[35] B. Liang, L. Zheng, Gesture recognition from one example using depth images, Lecture Notes on Software Engineering 1 (4) (2013) 339–343.
[36] H. Huang, Z. Ju, H. Liu, Real-time hand gesture feature extraction using depth data, in: International Conference on Machine Learning and Cybernetics (ICMLC), Vol. 1, IEEE, 2014, pp. 206–213.
[37] Z. Ren, J. Yuan, Z. Zhang, Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera, in: Proceedings of the 19th ACM international conference on Multimedia, ACM, 2011, pp. 1093–1096.
[38] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, IEEE, 2005, pp. 65–72.
[39] H.-M. Zhu, C.-M. Pun, Hand gesture recognition with motion tracking on spatial-temporal filtering, in: Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, ACM, 2011, pp. 273–278.
[40] J. Heikkila, Geometric camera calibration using circular control points, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (10) (2000) 1066–1077.
[41] J. A. Bilmes, et al., A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, International Computer Science Institute 4 (510) (1998) 1–248.
[42] Z. Ju, H. Liu, Fuzzy gaussian mixture models, Pattern Recognition 45 (3) (2012) 1146–1158.
[43] H. Farid, E. P. Simoncelli, Optimally rotation-equivariant directional derivative kernels, in: Computer Analysis of Images and Patterns, Springer, 1997, pp. 207–214.
[44] Aldebaran, https://www.aldebaran.com/en.
[45] M. Havlena, Š. Fojtu, D. Pruša, T. Pajdla, Towards robot localization and obstacle avoidance from nao camera, Tech. rep., Czech Technical University in Prague (2010).
[46] Z. Ju, D. Gao, J. Cao, H. Liu, A novel approach to extract hand gesture feature in depth images, Multimedia Tools and Applications (2015) 1–15.

Zhaojie Ju (M'08) received the B.S. in automatic control and the M.S. in intelligent robotics both from Huazhong University of Science and Technology, China, in 2005 and 2007 respectively, and the Ph.D. degree in intelligent robotics at the University of Portsmouth, UK, in 2010.

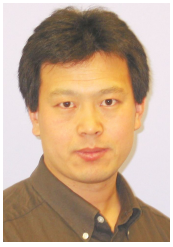
Dr Ju is currently a Senior Lecturer in the School of Computing, University of Portsmouth, UK. He previously held research appointments in the Department of Computer Science, University College London and School of Creative Technologies, University of Portsmouth, UK. His research interests are in machine intelligence, robot learning, pattern recognition and their applications in robotic/prosthetic hand control and human-robot interaction.

Xiaofei Ji received her M.S. and Ph.D. degrees from the Liaoning Shihua University and University of Portsmouth, in 2003 and 2010, respectively. From 2003 to 2012, she was the Lecturer at School of Automation of Shenyang Aerospace University. From 2013, she holds the position of Associate Professor at Shenyang Aerospace University. She has published over 40 technical research papers and 1 book. More than 20 research papers have been indexed by SCI/EI.

Her research interests include vision analysis and pattern recognition. She is the leader of National Natural Science Fund Project(Number: 61103123) and main group member of 6 National and Local Government Projects.



Jing Li received the Ph.D. degree from the Department of Electronic and Electrical Engineering at the University of Sheffield, U.K., in 2012. She is currently an Associate Professor with the School of Information Engineering at Nanchang University, China. Her research interests include content-based image retrieval, pattern recognition, and video analysis. She has authored or coauthored in various peer-reviewed journals, such as IEEE Transactions on Industrial Informatics, Information Sciences (Elsevier), etc.



Honghai Liu (M'02-SM'06) received his Ph.D degree in robotics from King's college London, UK, in 2003. He is currently a Professor of intelligent systems with the University of Portsmouth, Portsmouth, UK. He previously held research appointments at the Universities of London and Aberdeen, and project leader appointments in large-scale industrial control and system integration industry. He is interested in biomechatronics, pattern recognition, intelligent video analytics, intelligent robotics and their practical applications with an emphasis on approaches that could make contribution to the intelligent connection of perception to action using contextual information. He is Associate Editor of IEEE Transactions on Industrial Informatics, IEEE Transactions on Fuzzy Systems, IEEE Transactions on Human-Machine Systems.

Dr Liu is a Fellow of the Institution of Engineering and Technology.