Research Article

Advances in Mechanical Engineering

Advances in Mechanical Engineering
1–12

DOI: 10.1177/1687814014568488
aime.sagepub.com
SAGE

Three-dimensional pose estimation model for object with complex surface

Ronghua Li[1,2], Yong Chen[1], Changjiu Zhou[2] and Liandong Zhang[2]

Abstract

The proposed three-dimensional pose estimation model for object with complex surface, which primarily absorbs the essence of scale-invariant feature transform and iterative closest point algorithm, includes two steps, off-line and online. At first, two kinds of feature databases are established in the off-line operations. Then, the online process mainly has three steps. The first one is two-dimensional edge extraction from red–green–blue (RGB) information based on scale-invariant feature transform algorithm. The second one is three-dimensional surface reconstruction from the previous two-dimensional edge and the depth information obtained from depth camera. The last one is three-dimensional pose estimation based on camera calibration and iterative closest point algorithm. The Kinect camera is selected as the information acquisition device which can produce red–green–blue information and depth information. In the experiment, the container twist-lock with complex surface is taken as the object. The result shows that the accuracy of the proposed model is very high.

Keywords

Three-dimensional pose, scale-invariant feature transform algorithm, iterative closest point algorithm, twist-lock

Date received: 23 June 2014; accepted: 20 December 2014

Academic Editor: Xiaotun Qiu

Introduction

Three-dimensional (3D) pose estimation is one of the current important studies and is very useful in many fields. Shai Segal et al.[1] estimate the relative 3D pose of the cooperative satellites by on-board sensors to be used for the spacecraft formation flying or rendezvous and docking. Dan Lv et al.[2] achieve a 3D pose estimation model for the rigid objects on the ground to recognize the military vehicles automatically. Malik Saad Sultan et al.[3] propose a monocular camera vision system for a 6-degree-of-freedom (DOF) drawing robotic arm by estimating the 3D pose of the end effecter robustly. Furkan Kirac et al.[4] detect the 3D pose of hand gesture from single frame depth data to realize the human–computer interaction.

The research on 3D pose estimation belongs to a multi-interdisciplinary area, involving the projection geometry, digital image processing, computer graphics, computer vision, and many other disciplines.[5] The 3D reconstruction is the key technology of this research to restore the 3D space information of the objects through the basic elements (such as points, lines, and planes) from two-dimensional (2D) images. For achieving the quantitative analysis of the scale and positions of the objects, it also needs to study the relations between 3D coordinates of points, lines, and planes in 3D space and the corresponding ones in 2D images. The 3D

[1]School of Mechanical Engineering, Dalian Jiaotong University, Dalian, China
[2]Advanced Robotics and Intelligent Control Centre, Singapore Polytechnic, Singapore

Corresponding author:
Ronghua Li, School of Mechanical Engineering, Dalian Jiaotong University, Dalian 116028, China.
Email: lirh@djtu.edu.cn

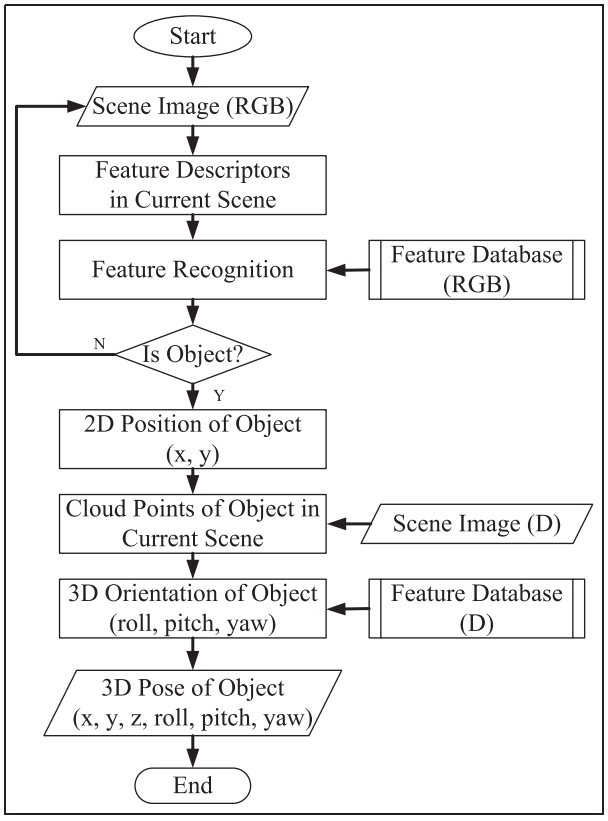

Figure 1. General architecture of proposed model.

information or 3D model is obtained by feature extraction, feature matching, reconstruction of key characteristics, triangulation, and data fusion.

The proposed model is investigated using Kinect camera produced by Microsoft Corporation. The software development kit (SDK) of the camera is open to the users.[6] Moreover, it has the ability to provide the depth information for every red–green–blue (RGB) pixel acquired. It is possible to build 3D point clouds using the depth information, which are very suited for 3D reconstruction and frame-to-frame alignment.[7–9] Furthermore, RGB data can be more appropriate for other processes such as loop closure detection.[10] Combining both characteristics seems to represent an opportunity to develop more on robotics navigation or object recognition.[11]

The general architecture of the model is shown in Figure 1, which primarily absorbs the essence of scale-invariant feature transform (SIFT) and iterative closest point (ICP) algorithm. It totally has two steps, off-line and online. At first, two feature databases are established to simulate the human memory in the off-line operations. Then, the online process mainly has three parts: (1) 2D edge extraction from RGB information based on SIFT algorithm; (2) 3D surface reconstruction from the previous 2D edge and depth information obtained from depth camera; and (3) 3D pose estimation based on camera calibration and ICP algorithm.

The SIFT descriptor proposed in Lowe[12] could keep the feature invariant after being rotated, resized, and even affined. The SIFT algorithm detects the feature in the scale space whose orientation is the principal direction of the neighborhood gradient. The SIFT algorithm has been continuously improved to achieve a higher stability.[13,14] The process of obtaining the SIFT descriptors includes five steps: detection of scale-space extrema, accurate key point localization, orientation assignment, generation of feature descriptors, and matching regulation.

ICP[15] is an algorithm employed to minimize the difference between two sets of point clouds. ICP is often used to reconstruct 2D or 3D surfaces from different scans and to localize robots and achieve optimal path planning (especially when wheel odometry is unreliable due to slippery terrain). The algorithm is conceptually simple and is commonly used in real time. It iteratively revises the transformation (translation, rotation) to minimize the distance between the points of two raw scans.

In the experiment, the container twist-lock with complex surface is taken as the object. The important significance is that the robotic technology is needed to replace the stevedores' heavy work to improve the efficiency and safety. At the same time, the twist-lock and corner casting together form a standardized rotating connector for securing shipping containers by coupling containers together during transferring. Each container needs at least four twist-locks, and there are many twist-locks to be used in the world. All of them are removed and installed by manual work now.

The rest of this article is organized as follows: the feature databases are established in off-line operations in section "Establishment of feature database in off-line operations"; section "2D edge extraction from RGB information based on SIFT algorithm" explains how to extract 2D edge from RGB information based on SIFT algorithm; in section "3D surface reconstruction from the 2D edge and depth information," 3D surface is reconstructed from the 2D edge and depth information; in section "3D pose estimation based on camera calibration and ICP algorithm," 3D pose parameters (*x, y, z, roll, pitch, yaw*) are computed based on the camera calibration and ICP algorithm; the experimental results are given in section "Experimental validations" to show the feasibility and performance of the proposed algorithm; finally, a brief conclusion and future work is presented in section "Conclusion and future work."

Establishment of feature database in off-line operations

Two kinds of feature databases are established in the off-line operations. One is the RGB feature database and the other is the point clouds database.

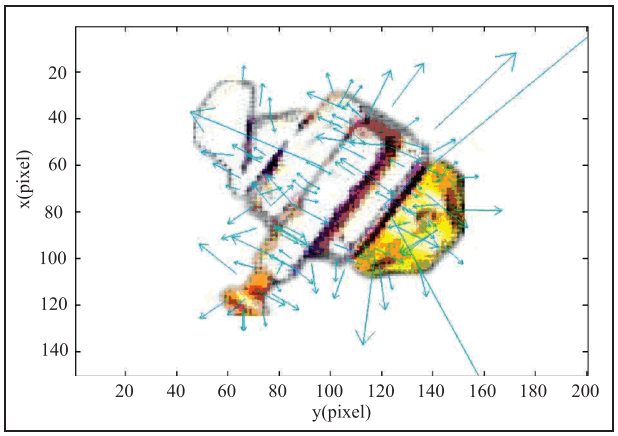

Figure 2. Descriptors of one feature image.

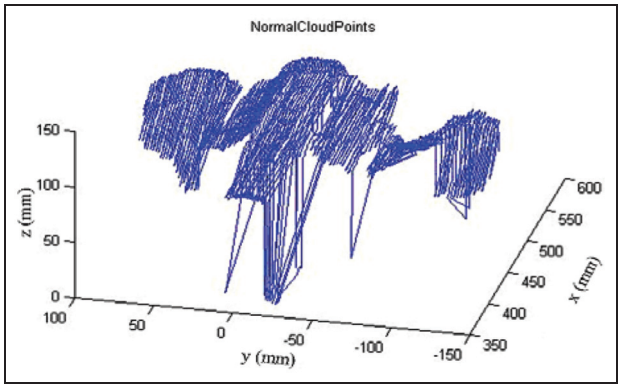

Figure 3. Point clouds database.

RGB feature database

The ideal RGB database is established to simulate the human memory as follows: first, the target is placed at the center of the regular polyhedron, and then, the images are acquired by some cameras, which are fixed on the centers of the polyhedron surfaces and while the optical axes of the cameras are in coincidence with the normal lines of the polyhedron. The more surfaces can make the object feature and the recognition performance better; however, it will make the algorithm more complex. Considering the complexity and precision, the octahedron is adopted in the experiment.

The eight images from different viewpoints need to be processed. Each feature image is described by two matrixes, which are the descriptors matrix and the location matrix, respectively. The descriptors matrix is a *K-by-128* matrix, where each row gives an invariant descriptor for one of the *K* key points. The descriptor is a vector of *128* values normalized to unit length. The location matrix is a *K-by-4* matrix, in which each row has the four values for a key point location (row, column, scale, and orientation). The establishment of feature database is off-line in order to save online resources. As shown in Figure 2, the descriptors of one feature image are displayed by arrows. The direction of the arrow represents the direction of the gradient in the position of the corresponding pixel; the length of the arrow represents the module of the gradient.

Point clouds database

The point clouds database is used in the ICP-based process. It is acquired in the standard state. At this time, the 3D pose parameters are $x = 500$, $y = 0$, $z = h$ (h is the highest point on the surface), $roll = 0$, $pitch = 0$, and $yaw = 0$, respectively, as shown in Figure 3.

2D edge extraction from RGB information based on SIFT algorithm

Detection of scale-space extrema

The scale-space theory is used to describe the multiscale characteristic of one image. The Gaussian Convolution Kernel is the only linear kernel to achieve the scale transform; therefore, a 2D scale space is defined

$$L(x, y, \sigma) = G(x, y, \sigma) \cdot I(x, y) \tag{1}$$

where $G(x, y, \sigma)$ is the invariable scale Gaussian function, (x, y) are the space coordinates, and σ is the scale level

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2 + y^2)/2\sigma^2} \tag{2}$$

The Difference of Gaussian (DoG) scale space is proposed in order to detect the stable critical point in the scale space effectively. The DoG is easy to be calculated, which is close to the scale-normalized Laplacian of Gaussian (LoG) operator

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) \cdot I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{3}$$

In order to find out the extreme points in scale space, each sampling point should compare with all the adjacent points. As shown in Figure 4, the middle detected point compares with the other adjacent points, which include the 8 ones in the same scale and the 18 ones in the adjacent scales.

Accurate key point localization

In order to enhance the stability and improve the noise immunity, the location and scales of the key points (to achieve sub-pixel accuracy) are accurately determined through fitting 3D quadratic function; meanwhile, the

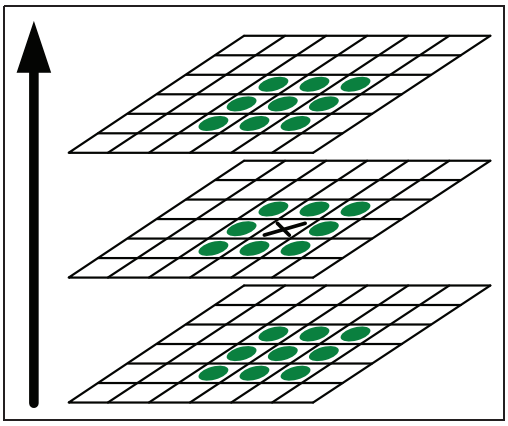

Figure 4. Sampling point comparing with all the adjacent points.

low-contrast points and instability edge points are removed.

The extreme value of badly defined *DoG* operator has a greater principal curvature across edge compared with a smaller curvature in the vertical edge direction. The principal curvature can be derived by a 2×2 Hessian matrix H. The derivatives in H are estimated by the differences of the adjacent sampling points.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{4}$$

The principal curvature of D is proportional to the eigenvalues of H. Let α be the largest eigenvalue and β the smallest eigenvalue, then

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{5}$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \tag{6}$$

Let $\alpha = r\beta$, then

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{7}$$

The value of $(r+1)^2/r$ is smallest when the two eigenvalues are equal. Therefore, in order to estimate whether the principal curvature is in a certain area r, the following inequality is just satisfied. In Lowe,[12] $r = 10$

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \tag{8}$$

Orientation assignment

The direction parameters are assigned to every critical point according to the distribution property of the neighborhood pixels' gradient direction to make the operators have the rotation invariance. The following formulas are the module and direction of the gradient at (x, y)

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{9}$$

$$\theta(x,y) = \tan^{-1}\frac{L(x,y+1) - L(x,y-1)}{L(x,y+1) - L(x-1,y)} \tag{10}$$

In practice, the sample process is carried out in the neighborhood window which takes the critical point as the center, while uses the histogram to count the gradient direction of the neighborhood pixels. The histogram range is 0°–360°, hereinto, every 10° are considered as a column. The peak of the histogram represents the principal direction of the neighborhood gradient at this critical point.

In the histogram, other directions, whose peak value is more than 80% of the principal peak value, are considered as the auxiliary directions. The use of auxiliary directions can enhance the robustness.

At this time, the key points have been detected completely. Each point has three parameters: position, scale, and direction.

Generation of feature descriptors

Above all, the coordinate axis rotates to the direction of the critical point to ensure the rotation invariance. Then, the 8×8 window is obtained, which takes the critical point as the center. As shown in Figure 5(a), the black point is the critical point, each grid represents a pixel, the direction of the arrow represents the direction of the gradient in the position of the corresponding pixel, the length of the arrow represents the module of the gradient, and the blue circle represents the Gaussian-weighted scope.

Afterward, the gradient histogram that has eight directions is calculated in each 4×4 block, as shown in Figure 5(b). The cumulative value for each gradient direction forms a seed point. In this method, a key point is expressed by four seed points, while every seed point has eight directions.

Match regulation

The SIFT feature descriptors of the twist-lock in scene and the eight feature images have been generated. Then, by the Euclidean distance we can obtain eight similarity values, which are the quantity of the matching points between the twist-lock and the feature database. The biggest value of them is effective. Simultaneously, the psychology threshold is set. If the

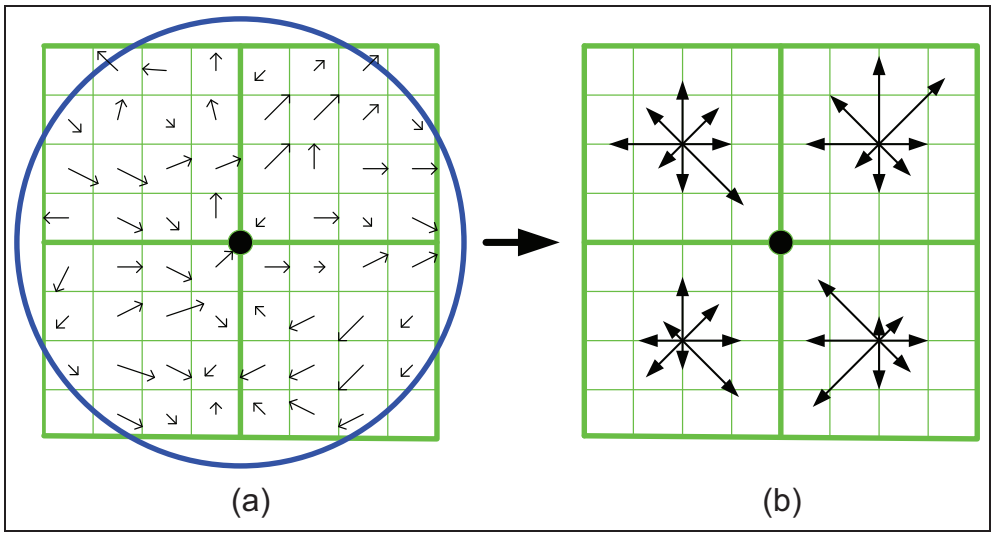

Figure 5. Feature descriptors. (a) Gradient of adjacent pixels. (b) Feature descriptors by gradient synthesis.

biggest value is greater than the psychology threshold, the region is considered as the corresponding object.

Experimental evaluation

The code is realized according to the above-mentioned model. A lot of experiments have been made to test the algorithm. As shown in Figure 6, the 2D edge can be extracted. The result indicates that processing accurately one color image digitized at 640×480 resolutions takes 200 ms online (Intel(R) Pentium(R) 4 CPU 3.00 GHz, 1.00 GB of RAM Physical Address Extension).

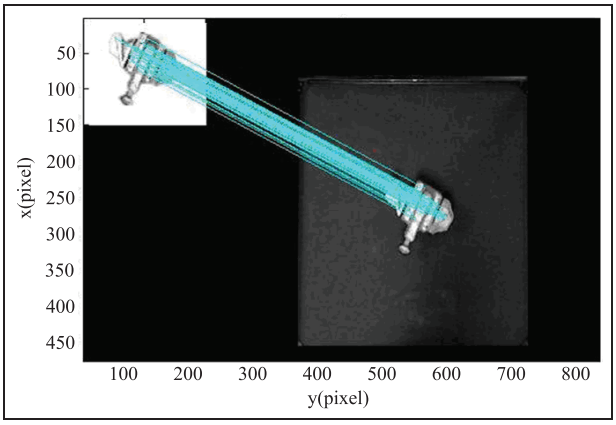

Figure 6. Result obtained by the SIFT algorithm.

3D surface reconstruction from the 2D edge and depth information

The 3D surface data can be obtained from the previous 2D edge and depth information. We can get the projective data in 3D coordinate (Figure 7(b)) from the original depth image (Figure 7(a)). In Figure 7(a), the bar on the right shows the pseudo-colors corresponding to the depth value, whose unit is millimeter. The zero level is the surface of the experiment table. We can see that the data are very noisy. How can we find 3D data of the object? We resort to the 2D edge (Figure 7(c)) for solution. After matching, the 3D surface can be reconstructed as shown in Figure 7(d).

3D pose estimation based on camera calibration and ICP algorithm

The 3D pose array includes three position parameters (x, y, z) and three orientation parameters (*roll, pitch, yaw*). As shown in Figure 8, O_o-$X_oY_oZ_o$ is the object coordinate and O_W-$X_WY_WZ_W$ is the world coordinate. On the basis of the previous result, the 3D pose parameters can be computed by camera calibration and ICP algorithm.

2D position by camera calibration

So far, the position of the twist-lock is obtained in the pixel coordinate system. After that, we need to transform this position into the real-world coordinate by camera calibration[16] (Figure 9). The transformation data from pixel coordinate to world coordinate are shown in Figure 10.

Orientation estimation based on ICP

Introduction of ICP. The ICP algorithm is given by Besl and McKay[15] in 1992. In general, ICP has to select some sets of points in one or both meshes and matches

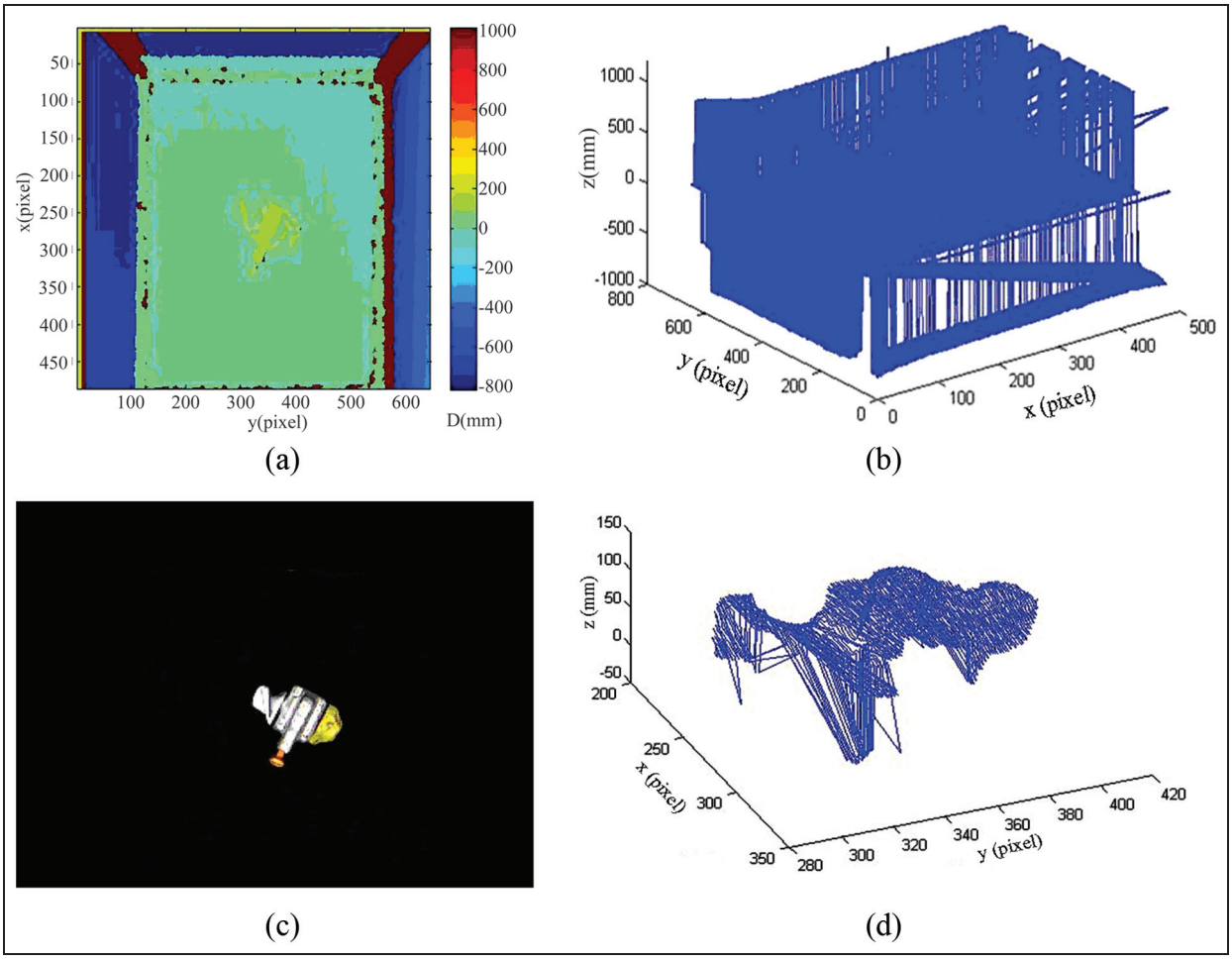

Figure 7. Three-dimensional (3D) surface reconstruction: (a) original depth image, (b) projective data in 3D coordinate, (c) 2D edge of object, (d) actual points cloud in pixel coordinate.

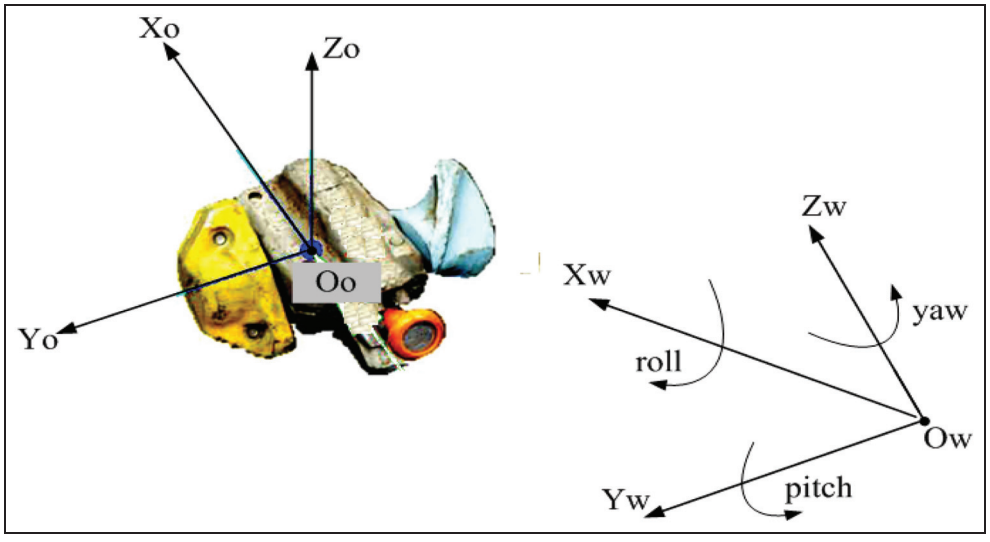

Figure 8. Three-dimensional (3D) pose definition.

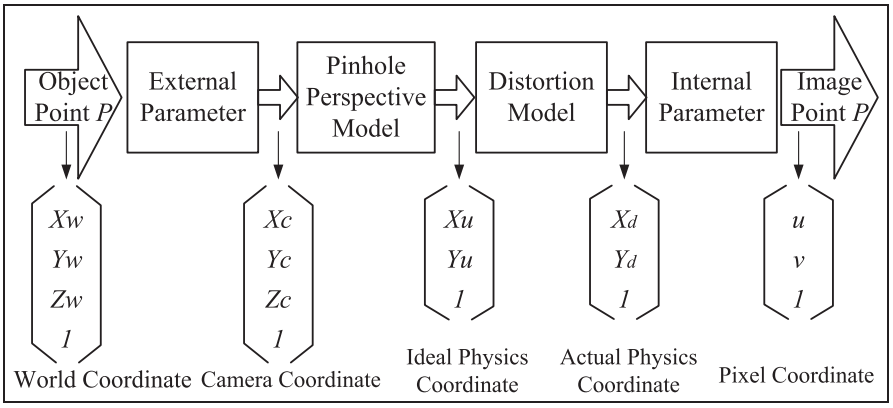

Figure 9. Camera calibration.

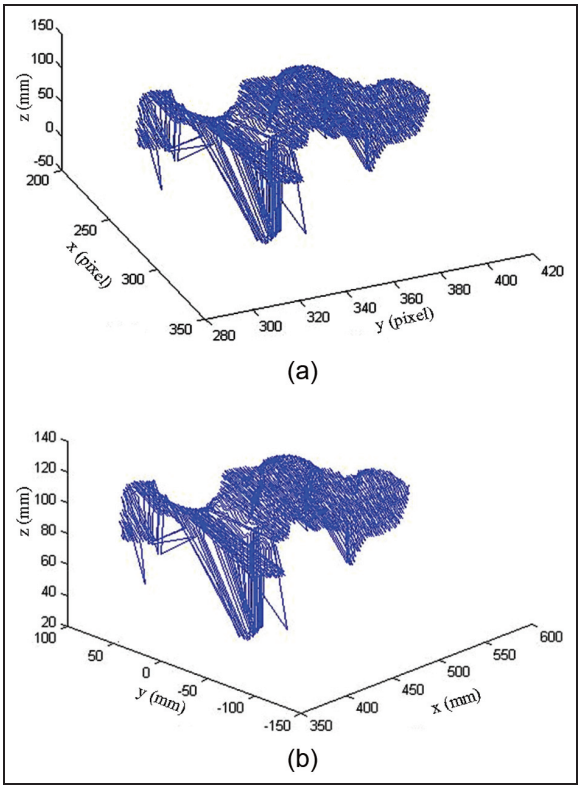

Figure 10. Coordinate transformation. (a) Actual points cloud in pixel coordinate. (b) Actual points cloud in world coordinate.

points to the samples in the other set. Meanwhile, a definition for errors is assigned and the errors are minimized by the iteration. The model shape can be a set of points, plotlines, parametric curves, implicit curves, or implicit surfaces.

ICP algorithm moves a data shape P to align with a model shape T. Let Q be the resulting set of points and c be the closet point operator

$$Q = c(P, T) \tag{11}$$

Given the result set Q, which is the correspondence of P and T, the least-squares registration is computed by

$$(H, d) = q(P, Q) \tag{12}$$

Transform every point of P by matrix H

$$H(P) = R(P) + T \tag{13}$$

A set of points S from the data shape to the model shape T is given. Let S and T represent source and target data shape, respectively. Iteration is initialized with $P_0 = S$, $R_0 = I$, $T_0 = 0$, and $k = 0$; essentially, the algorithm steps are as follows:

Step 1: compute the closest points by equation (11)

$$Q_k = c(P_k, T) \tag{14}$$

Step 2: estimate the transformation parameters using a mean-square cost function by equation (12)

$$H_k = \begin{bmatrix} R_k & T_k \\ 0 & 1 \end{bmatrix} \tag{15}$$

and make the distance d_k between an individual data point P_k and model shape T.

Step 3: transform the points using the estimated parameters by equation (13)

$$P_{k+1} = H_k(P_0) = R_k P_0 + T_k \tag{16}$$

Step 4: terminate the iteration when the change in mean-square error falls below a threshold $|d_k - d_{k+1}| < \tau$.

Given 3D shape from point clouds database and a 3D shape in the actual state, the ICP will get the optimal solution if the "actual data" set is the subset of the "point clouds database." However, a copy of a 3D

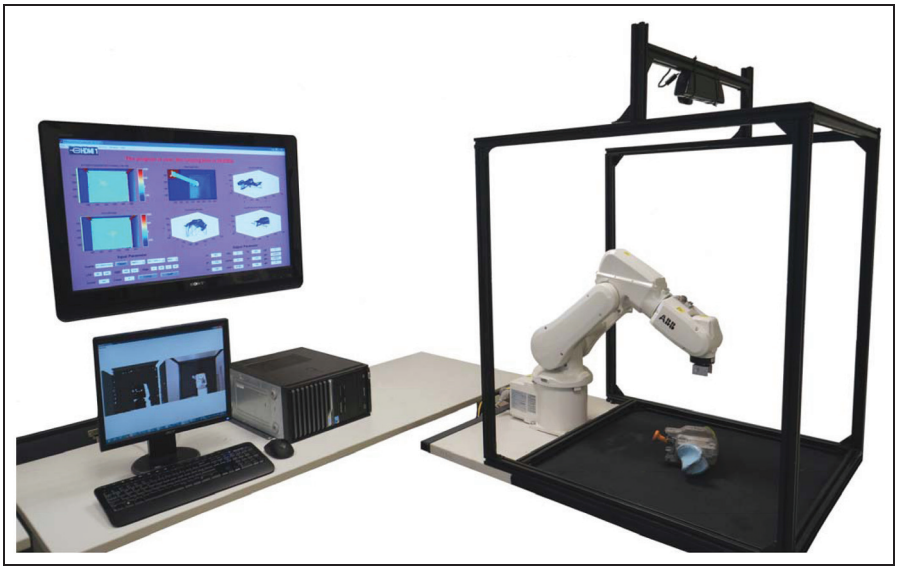

Figure 11. Experimental platform.

parametric spine is rotated to be difficult to match the reference if the 3D shapes are built from different directions, which only partially overlap between each other. As a result of that, the initial relative pose estimation should not be too different from the real one.

Orientation estimation. We can obtain the refined transformation (rotation matrix and translation matrix) from point clouds database to 3D shape in actual state.

The *yaw, pitch*, and *roll* rotations can be used to place a 3D body in any orientation. A single rotation matrix can be formed by multiplying the *yaw, pitch*, and *roll* rotation matrices to obtain

$$R(\alpha,\beta,\gamma) = R_z(\alpha)R_y(\beta)R_x(\gamma) = \begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{pmatrix} \tag{17}$$

Suppose an arbitrary rotation matrix

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{18}$$

is given. By setting each entry equal to its corresponding entry in equation (17), the equations can be solved for α, β, and γ. Note that $r_{21}/r_{11} = \tan\alpha$, $r_{32}/r_{33} = \tan\gamma$, $r_{31} = -\sin\beta$, and $\sqrt{r_{32}^2 + r_{33}^2} = \cos\beta$. Solving for each angle yields

$$yaw = \alpha = \tan^{-1}\left(\frac{r_{21}}{r_{11}}\right) \tag{19}$$

$$pitch = \beta = \tan^{-1}\left(\frac{-r_{31}}{\sqrt{{r_{32}}^2 + {r_{33}}^2}}\right) \tag{20}$$

$$roll = \gamma = \tan^{-1}\left(\frac{r_{32}}{r_{33}}\right) \tag{21}$$

Experimental validations

Experimental platform and software

The experimental platform in Figure 11 is composed of one experiment table with the steadier apparatus, one main machine and expanded monitor, one Kinect camera, and one ABB robot. The 3D pose estimation software, which is developed by our group, can integrate all the previous algorithms. The interface of the 3D pose measurement software is as shown in Figure 12.

Experimental result

The main experimental flow is published to carry on the analysis, as shown in Figure 13. The object to be searched for is a twist-lock.

Figure 14 shows the RGB-D images in the typical scenes. The first scene is that there is only the smaller rotation of z (*yaw*), whose direction is

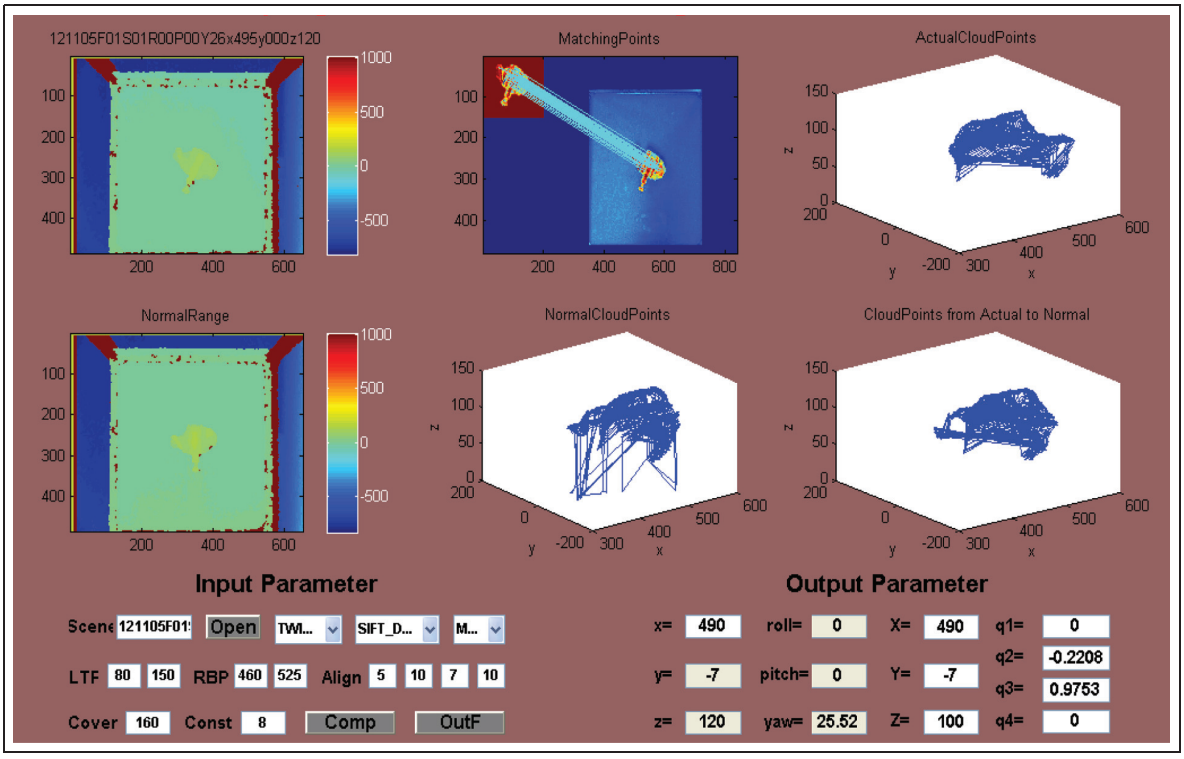

Figure 12. 3D pose estimation software.

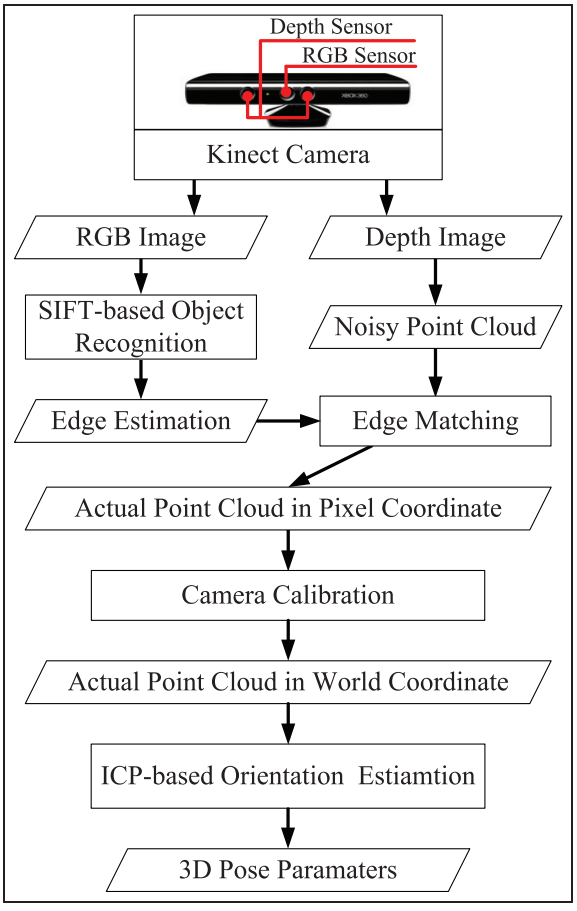

Figure 13. Main experimental flow.

positive. There is also only the rotation of z (*yaw*) in the second and third scenes; however, the rotation is larger in the second scene and the direction is negative in the third scene. Meanwhile, there is only the rotation of y (*pitch*) in the fourth scene, and there are the rotations of x (*roll*) and y (*pitch*) in the fifth scene. Moreover, the rotations of three axes exist in the sixth scene.

The matched experimental results are shown in Table 1. "T" represents the true value, "M" represents the measured value, and "E" represents the error value. The unit of x, y, and z is millimeter, and the unit of *roll, pitch, yaw* is degrees. All of them denote that the accuracy of the proposed model is very high.

The result indicates that processing accurately one scene takes about 1.5 s online (Intel(R) Pentium(R) 4 CPU 3.00 GHz, 1.00 GB of RAM Physical Address Extension). Since the twist-lock to be recognized is stationary in the actual application, this speed of the proposed model can meet the requirements of the robot grasping very well.

Conclusion and future work

Inspired by the SIFT and ICP algorithm, a 3D pose estimation model is proposed for object with complex surface, which includes both the off-line step and the online step. The off-line step includes the RGB database

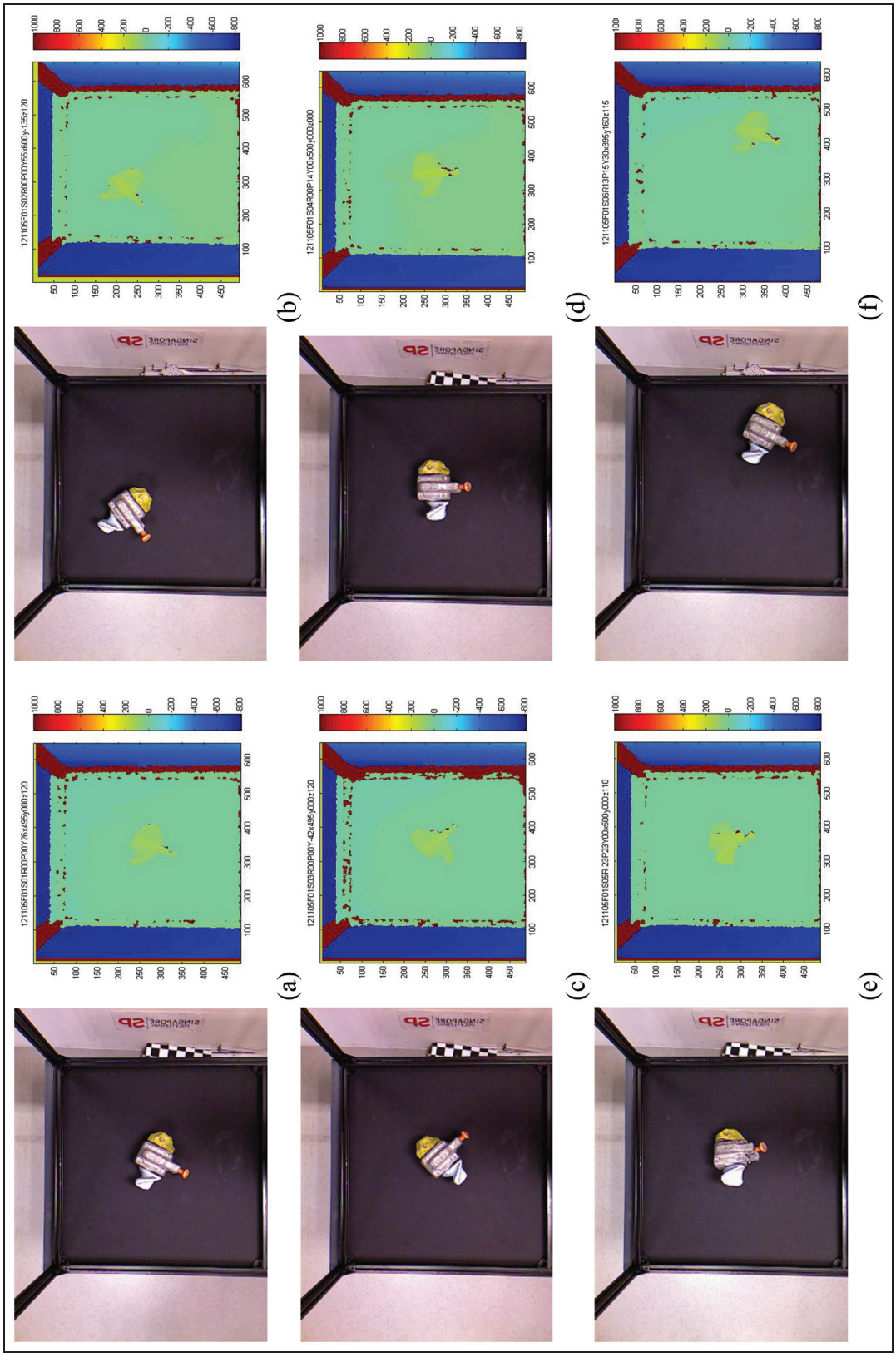

Figure 14. RGB-D images in the experiments: (a) RGB-D images in first scene, (b) RGB-D images in second scene, (c) RGB-D images in third scene, (d) RGB-D images in fourth scene, (e) RGB-D images in fifth scene, and (f) RGB-D images in sixth scene.

Table. 1. Experimental result.

Scene No.	First			Second			Third			Fourth			Fifth			Sixth		
Data type	T	M	E	T	M	E	T	M	E	T	M	E	T	M	E	T	M	E
x (mm)	495	490	−5	600	592	−8	495	490	−5	500	505	5	500	498	−2	395	384	−11
y (mm)	0	−7	−7	−135	−133	2	0	1	1	0	−6	−6	0	6	6	160	166	6
z (mm)	120	120	0	120	120	0	120	120	0	120	120	0	110	109	−1	115	118	3
Roll (°)	0	0	0	0	0	0	0	0	0	0	0	0	−23	−25.01	−2.01	13	10.73	−2.27
Pitch (°)	0	0	0	0	0	0	0	0	0	14	13.61	−0.39	23	20.26	−2.74	15	12.01	−2.99
Yaw (°)	26	26.31	0.31	55	51.77	−3.23	−42	−40.78	1.22	0	0	0	0	0	0	30	27.27	−2.73

used for SIFT and the point clouds database used for ICP. The online process mainly has three steps, which are 2D edge extraction from RGB information based on SIFT algorithm, 3D surface reconstruction from the previous 2D edge and the depth information obtained from depth camera, and 3D pose estimation based on camera calibration and ICP algorithm. The Kinect camera and the container twist-lock are selected, respectively, as the information acquisition device and the recognized object. The result shows that the accuracy is very high.

Inheriting this article's work, the future work will concentrate on the cooperation with gripping robot. Moreover, the ICP used in this article is time-consuming, so the speed of the algorithm needs to be improved to meet the requirements of grasping a dynamic object and then to achieve a real-time system.

Declaration of conflicting interests

The authors declare that there is no conflict of interest.

Funding

This work was supported by Scientific Research Project of the Education Department in Liaoning Province (No. L2013198) and National Nature Science Foundation of China (No. 51305055).

References

1. Segal S, Carmi A and Gurfil P. Stereovision-based estimation of relative dynamics between noncooperative satellites: theory and experiments. *IEEE Trans Contr Syst Technol* 2014; 22(2): 568–584.
2. Lv D, Sun J, Li Q, et al. 3D pose estimation of ground rigid target based on ladar range image. *Appl Optic* 2013; 52(33): 8073–8081.
3. Sultan MS, Chen X, Ma G, et al. Hand-eye 3D pose estimation for a drawing robot. In: *IEEE international conference on mechatronics and automation*, Takamatsu, Japan, 4–7 August, 2013, pp.1325–1331. Takamatsu: IEEE Computer Society.
4. Kirac F, Kara YE and Akarun L. Hierarchically constrained 3D hand pose estimation using regression forests from single frame depth data. *Pattern Recog Lett* 2014; 50: 91–100.
5. Newman P, Sibley G, Smith M, et al. Navigating, recognizing and describing urban spaces with vision and lasers. *Int J Robot Res* 2009; 28(11–12): 1406–1433.
6. Microsoft Corporation. http://www.microsoft.com/en-us/kinectforwindows/
7. Yao Y and Fu Y. Contour model-based hand-gesture recognition using the Kinect sensor. *IEEE Trans Circ Syst Video Technol* 2014; 24(11): 1935–1944.
8. Mastorakis G and Makris D. Fall detection system using Kinect's infrared sensor. *J Real-Time Image Process* 2014; 9: 635–646.
9. Chan W, Yue H, Wu X, et al. Real-time obstacle detection for legged robot using the Kinect sensor. *Adv Robot* 2014; 28(20): 1375–1387.

10. Konolige K and Agrawal M. FrameSLAM: from bundle adjustment to real-time visual mapping. *IEEE Trans Robot* 2008; 24(5): 1066–1077.
11. Perez Bonnal E. *3D mapping of indoor environments using RGB-D Kinect camera for robotic mobile application*. Dissertation, Politecnico di Torino, Italy, 2011.
12. Lowe DG. Object recognition from local scale-invariant features. In: *International conference on computer vision*, Corfu, Greece, September, 1999, pp.1–8. New York: IEEE Press.
13. Lowe DG. Distinctive image features from scale-invariant keypoints. *Int J Comput Vision* 2004; 60(2): 91–110.
14. Ke Y and Sukthankar R. PCA-SIFT: a more distinctive representation for local image descriptors. In: *Proceedings of the IEEE Computer Society conference on computer vision and pattern recognition*, Washington, DC, July, 2004, pp.511–517. Washington, DC: IEEE Computer Society.
15. Besl PJ and McKay ND. A method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell* 1992; 14(2): 239–256.
16. Wang Y, Wei Z and Shao M. A new method to calibrate robot visual measurement system. *Adv Mech Eng* 2013; 2013: 1–8.