Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 8252015, 9 pages
http://dx.doi.org/10.1155/2016/8252015

Research Article

Kinect-Based Vision System of Mine Rescue Robot for Low Illuminous Environment

Wanfeng Shang, Xiangang Cao, Hongwei Ma, Hailong Zang, and Pengkang Wei

School of Mechanical Engineering, Xi'an University of Science & Technology, Xi'an, Shaanxi 710054, China

Correspondence should be addressed to Wanfeng Shang; shangwanfeng@gmail.com

Received 25 December 2014; Accepted 4 February 2015

Academic Editor: Yong Zhang

This paper presents Kinect-based vision system of mine rescue robot working in illuminous underground environment. The somatosensory system of Kinect is used to realize the hand gesture recognition involving static hand gesture and action. A K-curvature based convex detection method is proposed to fit the hand contour with polygon. In addition, the hand action is completed by using the NiTE library with the framework of hand gesture recognition. In addition, the proposed method is compared with BP neural network and template matching. Furthermore, taking advantage of the information of the depth map, the interface of hand gesture recognition is established for human machine interaction of rescue robot. Experimental results verify the effectiveness of Kinect-based vision system as a feasible and alternative technology for HMI of mine rescue robot.

1. Introduction

Rescue is very urgent after coal mine accident because after 48 hours, victim mortality drastically increases owing to exposure to bad air and lack of food, water, and medical treatment [1, 2]. The mine tunnel may further collapse injuring or killing the trapped survivors or rescuers. Noxious gas leaks and explosions are also possible. At such time, robots have a great potential to assist in these underground operations ahead of rescue teams and report conditions that may be hazardous to people. When explosive conditions exist or heavy smoke enters the mine roadway, robots can become an invaluable tool. These mobile robots navigating deep into rubbles can search for survivors and transfer on-site video and atmospheric monitoring information for rescuers to confirm safe state or identify potentially hazardous conditions for mine rescue team [3].

It is an urgent challenge for intelligent rescue robots working underground mine tunnel that is a long distance from ground. After mine accidents occur, the environment of underground mine tunnel is unknown and low illuminous and even has no light (the power system has collapsed). With the development of mobile devices and sensors, voice and hand gestures have become a popular way to interact with robots and personal computers [4]. Voice is vulnerable to environmental noise and is not suitable for continuous remote control. It is difficult to separate different people's intonation. Further, survivors trapped under ground may not easily take a breath for exiting harmful gas. So the voice data is unfit for human machine interaction (HMI) underground mine tunnel. In comparison with voice, gestures are more natural, intuitive, and robust. Besides, gestures can convey more extensive amount of information than sound. Thus, many researchers pay attention to gesture recognition research.

Currently, most technologies of the hand gesture recognition are based on ordinary cameras [5]. However, RGB color images from the ordinary camera have a disadvantage that they cannot provide enough information for tracking hands in 3D space because much of the spatial position information has to be inferred from 2D-3D mappings. During the imaging process, a lot of information is lost. In recent years, with the development of somatosensory interactive devices, the research methods based on the depth camera are rising. The camera is capable of obtaining the 3D depth information, such as TOF camera [6], stereophotogrammetric system [7]. Both of them are very accurate, but their use is limited by high cost, time, space, and expertise requirements. Kinect provides a possibility for improving the tradeoff between performance and price in the design of gesture interfaces.

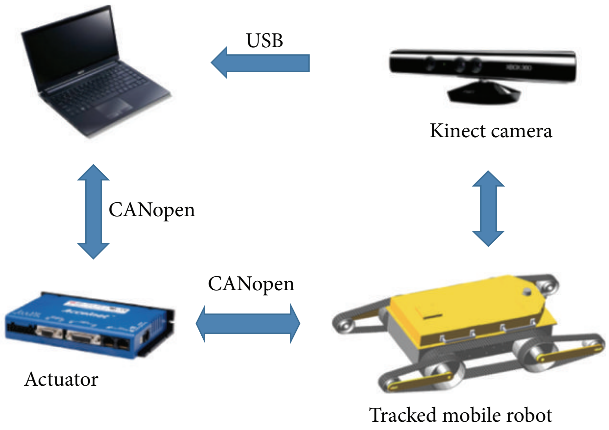


Figure 1: Architecture of vision system of rescue robot.

Kinect is a new integrated 3D sensor capturing depth image and RGB image. It has been widely used in many different fields, such as entertainment [8], industry automation [9], medical applications [10], and remote control of robot [11], because of its inexpensive depth sensing. The complementary nature of the depth and visual RGB information in Kinect sensor opens up new opportunities to solve fundamental problems in human behavior recognition for HMI. On the condition that the recognition environment is very dark, the target recognition will be conventionally out of function. While a depth image captured by Kinect is not affected by lights, the target recognition cannot be affected by complex background or lights in the field of object recognition. Hence, the target or hand gesture can be recognized in the dark or in places of complex images with Kinect. Furthermore, the ordinary camera is unfit for working underground where its image may lose detailed information. It is because the environment of underground mine tunnel is low illuminous, even dark in part of tunnel. Therefore, the Kinect is a viable sensor to use for HMI of rescue robot underground mining tunnel.

However, few researches are about the application of Kinect to a deployed rescue robot, though the Kinect has been used extensively in robot for navigation [12], environment mapping [13], and object manipulation [14]. One undergraduate team from the University of Warwick installed Kinect on robot to perform 3D SLAM in simulated disaster settings, namely, the RoboCup Rescue competition [15]. But the environment is indoors, and moreover the highly structured RoboCup course (made largely of diffuse plywood) lends itself easily to the task. Suarez and Murphy [1] reviewed Kinect's use in rescue robotics and similar applications and highlighted the associated challenges but did not detail the Kinect operation technology on rescue robot for hand gesture, navigation, remote control, and so forth. Development of Kinect makes some classical recognition methods and also introduces some new research technologies to improve human behavior recognition. So far Kinect-based full-body 3D motion estimation has been exploited to track body joints [8–10], while accurately recognizing small gestures, in particular hand motions, is still hard work. Many researches mainly include hand gesture and body parts recognition and posture tracking. Raheja et al. [16] used Kinect to track fingertip and palm via its SDK apps. Thanh et al. [17] team captured 3D histogram data of finger joint information from Kinect images and recognized hand gesture with TF-IDF algorithm. Doliotis et al. [18] proposed a clutter-tolerant hand segmentation algorithm where 3D pose estimation was formulated as a retrieval problem, but the local regressions for smoothing the sequence of hand segmentation widths did not allow the algorithm to achieve real-time performance. With the aim to extend the set of recognizable gestures, Wan et al. [19] proposed a method to recognize five gestures using Kinect, but one limitation of this work is the impossibility to detect vertical gestures. Pedersoli et al. [20] displayed a unified open-source framework for real-time recognition of both static hand-poses and dynamic hand gestures, but relying only on depth information. Based on the above, it can be attained that hand gesture recognition depends more on depth map, while body posture is liable to focus on body skeleton joint data captured from Kinect.

Therefore, the paper proposes a Kinect-based vision system for rescue robot underground mining tunnel and achieves hand gesture recognition combining depth map and skeleton joint information.

This paper is organized as follows. Section 2 introduces the architecture of Kinect-based vision system of rescue robot. Section 3 presents the hand gesture recognition in low illuminous environment using OpenNI and NiTE library for Windows. The experimental results are analyzed in Section 4. Finally, the conclusion and some future work are given in Section 5.

2. Architecture of Kinect-Based Vision System

The Kinect-based vision system architecture of rescue robot for gesture recognition is shown in Figure 1. The system is composed of motor actuators, Kinect, PC, and a platform of tracked mobile robot used underground mine tunnel. Kinect, directly connected to host computer through USB port, is actually a 3D somatosensory camera with a low price. On the Kinect, there is an Infrared (IR) Projector, a Color (RGB)

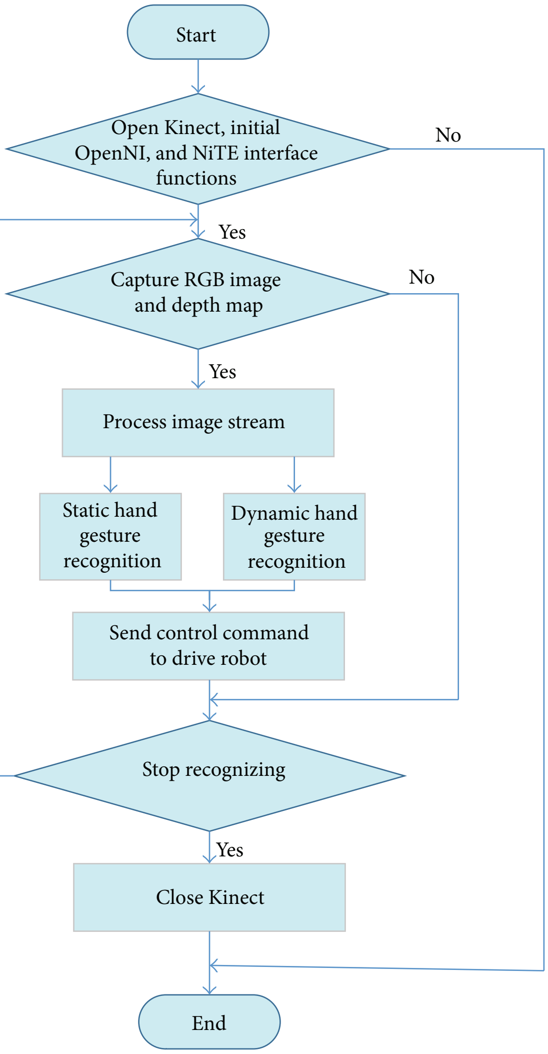

Start
Open Kinect, initial OpenNI, and NiTE interface functions
No
Yes
Capture RGB image and depth map
No
Yes
Process image stream
Static hand gesture recognition
Dynamic hand gesture recognition
Send control command to drive robot
Stop recognizing
Yes
Close Kinect
End

Figure 2: Hand gesture recognition workflow.

Camera, and an Infrared (IR) Sensor. For purposes of 3D sensing, the IR Projector emits a grid of IR light in front of it. This light then reflects off objects in its path and is reflected back to the IR Sensor. The pattern received by the IR Sensor is then decoded in the Kinect to determine the depth information and then is sent to another device via USB for further processing. It offers 640 ∗ 480 pixels' (about 30 fps) RGB image and depth image. Each RGB image is 24 bits. Each pixel in the depth image is 16 bits/mm, whose effective depth ranges from 0 mm to 4096 mm. Kinect software is capable of automatically calibrating the sensor based on the user's physical environment, accommodating for the presence of obstacles.

The system in Figure 1 is a slave control model with Kinect-based vision system. The computer acquires depth information of hand gesture from survivors, gives recognition results, and then transmits commands to actuators to control robots for serving survivors or other operators. It mainly serves survivors tapped in collapsed mining tunnel. When the gas in underground tunnel is heavy smoke and too bad for these survivors to breath, they cannot make any sound. At this time, the only way to control rescue robot is hand gesture recognition of HMI for survivors.

3. Hand Gesture Recognition

Generally, hand gestures fall into two categories such as static gestures and dynamic gestures. Both of them require proper recognition means by which they can be properly defined to the machine. Since OpenNI and NiTE library own the framework of Kinect, this paper used these library functions to realize the simple hand action, while being more focused on recognizing details about static hand gesture. The workflow is showed by Figure 2.

3.1. Capture Depth Image. For the hand gesture recognition, the first step is to obtain the image data captured from Kinect. With the OpenNI and NiTE, we can obtain the images in 640 ∗ 480 pixels' resolution at 30 fps. The OpenNI standard API enables natural-interaction developers to track real-life (3D) scenes by utilizing data calculated from the input of a sensor, for example, representation of a hand location. OpenNI is an open-source API that is publicly available.

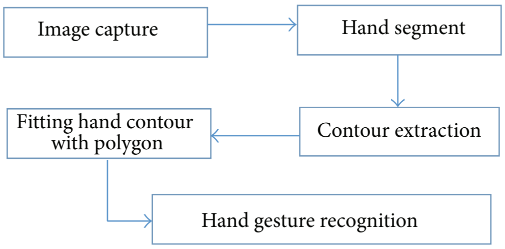


Figure 3: Workflow of static hand gesture recognition.

In this project, the Kinect is driven by the program of OpenNI, and the usage of NiTE library is based on the driver. The depth map needs to be captured by NiTE class. Hence, there must be three steps involving initialization OpenNI, Open device, and initialization NiTE to open Kinect. Before using OpenNI classes and functions, the necessary step is to initialize OpenNI, and then we can acquire the device information by defining the object of a device class and using an open function. When using NiTE classes and functions, it is similar to OpenNI; namely, it needs to be initialized in defined function. Through the above steps, Kinect can be used normally.

NiTE includes hand tracking framework and gesture recognition framework which is completed on the basis of the depth data. The method of obtaining the depth data is encapsulated in the HandTracker class, and the class provides methods to get hand position, transform coordinate, and detect hand gesture. The HandTrackerRef class stores the hand ID and the recognition results. Finally, the depth information captured needs to be converted into a gray scale image.

3.2. Static Hand Gesture Recognition. In above section, we have obtained the depth map and RGB data about hand gestures. In the next, we need to recognize the fingertip and count the number based on the hand tracking data. The Kinect with the released NiTE framework can provide 20 major joints of human skeleton, but only one joint is about palm information to locate a hand. The palm joint does not include the contour of the palm, fingers, and fingertips information and then not identify the static gestures or fingers. Hence, the paper proposes fingertip recognition based on extraction contour of straight and bend finger. The workflow of static hand gesture recognition is shown in Figure 3. The process is to first locate and segment a hand from the depth streams and then to extract contour and detection fingertip using K-curvature algorithm.

3.2.1. Hand Segment. To get the contour of hand, we need to segment hand from the depth streams. In this project, we use the segmentation method based on the depth thresholding. The segmentation range is from the depth value of a hand center point to the value adding or subtracting the depth threshold.

Let us define the hand skeleton joint as C_i, i = left and right, respectively, representing the palm center point of left hand and right hand. We can easily get C_i from depth image of hand gesture using open-source function of Kinect SDK software. The distance between the hand center point C_i and Kinect d_i can be expressed by

$$d_i = \text{frame} \cdot \text{width} \times C_i \cdot Y + C_i \cdot X, \qquad (1)$$

where frame represents a framework of depth map and width denotes a width value of depth image. Each pixel in depth image is 16 bits, but only 13 bits is effective for depth value. So, the true depth of each pixel depth$[j]$ can be achieved by

$$d_j = \text{depth}\,[j] \gg 3. \qquad (2)$$

The judging rule for hand segment is defined as

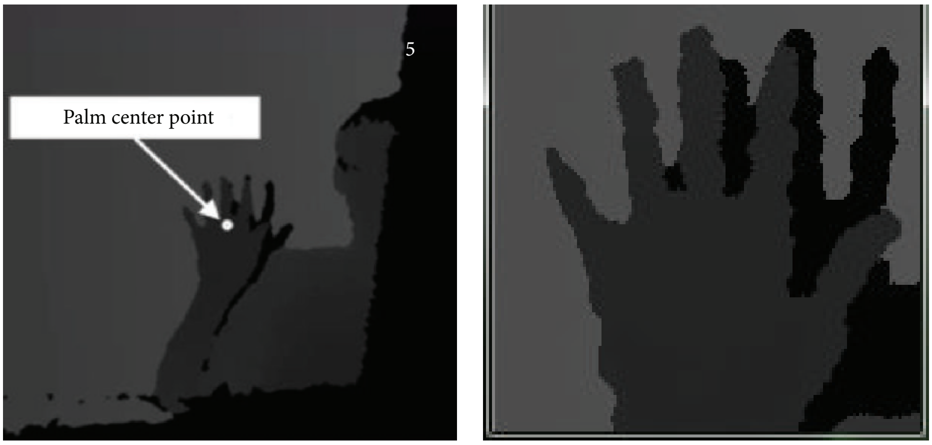
$$sd = \begin{cases} 1 & d_j \le d_i + s \\ -1 & d_j > d_i + s, \end{cases} \qquad (3)$$

where s is a little constant denoting a depth threshold and sd is a logic result to judge whether each pixel point of depth image is within the palm. If each pixel value in a depth image is less than the sum of the depth of skeleton joint and a depth threshold value, the pixel point is within the palm; otherwise it is not within the palm. In addition, if $s < 0$ the threshold value s has a change range, which can be estimated according to length of bend finger. According to (1)–(3), we get the segment result of one of hand gestures, as shown by Figure 4.

In fact, there is no absolute static state and hand gesture is generally dynamic, and we need the recognition system to respond fast to further remote control or navigation or complex HMI. Thus the recognition program of hand gesture must run fast. The method provides a feasibility of a simple and fast operation to ensure the accuracy of the dynamic hand segmentation.

3.2.2. Hand Contour Extraction. Before extracting the hand contour, we have to improve the image further. In Figures 5 and 6, due to the RGB camera and the depth camera not locating at the same position, their pixel coordinates of a corresponding object in RGB image and depth image cannot be matched. There exist some shadows around the hand, which can severely affect the performance of the recognition. Besides, there is inevitable noise in the images. Thus we must preprocess the images to eliminate them. This paper applies the median filter method [20] to eliminate the noise and uses the image binarization method to extract the hand shape, as indicated in Figure 5. Due to the difference of gray scale between black and white, the hand contour is very distinct. We can use the open-source function of OpenNI about edge process to effectively extract the hand contour, as displayed in Figure 6.

3.2.3. Fitting Hand Contour with Polygons. For a detailed recognition of hand gesture, the basic idea is to fit the hand contour with polygons and then count the hand finger by identifying the convex points and concave points between fingers. This paper uses K-curvature algorithm to recognize fingertips, that is, convex points and concave points of palm. As shown in Figure 7, the point C is the palm center point,



(a) Depth image

(b) Hand segmentation

Figure 4: Hand segmentation result.

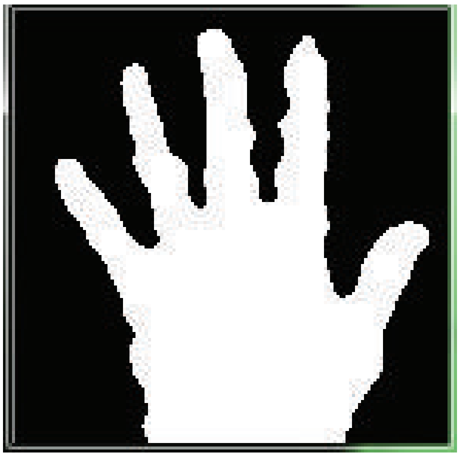


Figure 5: Image binarization.

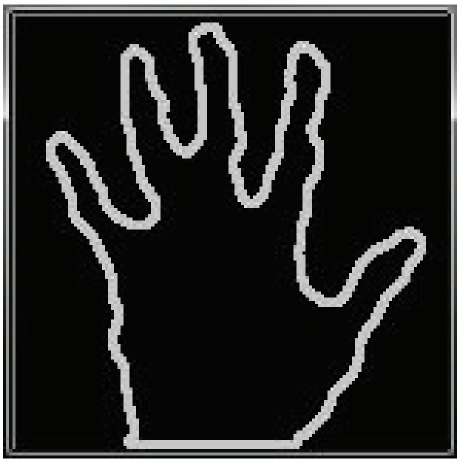


Figure 6: Hand contour extracted.

$P(i)$ is any point of the hand contour, $P(i+k)$ is the Kth point after $P(i)$, and $P(i-k)$ is the Kth point before $P(i)$. $\vec{V}_1$ is a vector that points from $P(i+k)$ to $P(i)$. $\vec{V}_2$ is a vector that points from $P(i-k)$ to $P(i)$. We take a cosine function of the angle α between $\vec{V}_1$ and $\vec{V}_1$ as K-curvature of $P(i)$ point. The angle α can be expressed by

$$\cos\alpha = \frac{\vec{V}_1 \cdot \vec{V}_2}{\left|\vec{V}_1 \cdot \vec{V}_2\right|}. \tag{4}$$

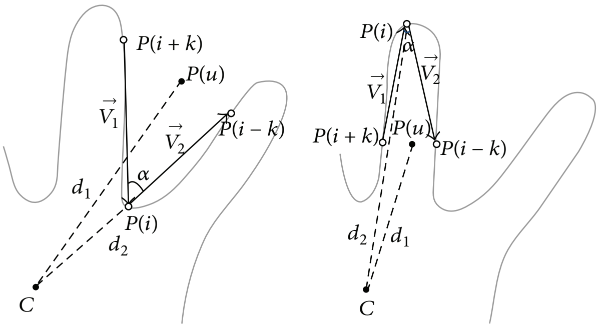
By judging whether the angle α is within a certain angle, we can further decide whether the $P(i)$ point in the hand contour is convex or concave points for a finger. It is important to choose a proper angle threshold to judge the α. If the angle threshold is too large, the wrist neighborhood may be misunderstood for fingertip, while a too little threshold will cause recognization failure. This paper used the angle threshold less than 55 degrees that was verified many times. If the α is less than the defined angle threshold, the $P(i)$ in the hand contour is a convex or concave point of a palm. Assume d_1 is a Euclidean distance between the middle point on the line $P(i+k)$ $P(i-k)$ and the point C, and d_2 is a Euclidean distance between the $P(i)$ and the point C. We define the rules as follows.

If $d_1 > d_2$, $P(i)$ is a concave point between fingers.

If $d_1 < d_2$, $P(i)$ is a convex point of finger, that is, fingertip.

Using the above defined law, the recognition result of hand polygon contour is showed by Figure 8. In the image, the blue line presents the hand contour, the red circles indicate the fingertips (namely, convex points), and the blue points show the concave points. The bigger red circle is the central point of the contour, and it is calculated by the following equations:

$$\begin{gathered} m_{ij} = \sum_{x,y} \left(\text{arry}\,(x,y) \cdot x^i \cdot y^j \right), \\ \overline{x} = \frac{m_{10}}{m_{00}}, \qquad \overline{y} = \frac{m_{01}}{m_{00}}. \end{gathered} \tag{5}$$



(a) Hand concave point (b) Hand convex point

FIGURE 7: Hand polygon contour vertex recognition.

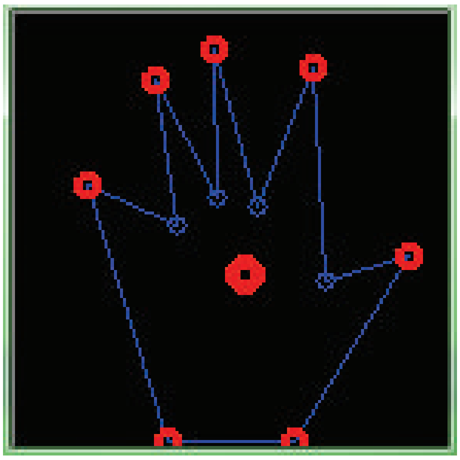


FIGURE 8: Polygon fitting of hand gesture.

However, for a practical application, the method cannot quite express all numbers of hand gestures only to count the above point, especially in the situation that the depth map captured by Kinect is dithering. To solve the problem, the paper adds an auxiliary point which locates at the point away from the contour central point 50 pixels and then counts the number of the convex points above the auxiliary point. The recognition results are shown in Tables 1 and 2. From the two tables, we can determine the criterion that suits to each number and then recognize the number via the criterion.

3.3. Dynamic Hand Gesture Recognition. For dynamic hand gesture recognition, this paper used dynamic time warping (DTW) [21] to detect hands in complex and cluttered background. DTW is dynamic programming and needs a lot of sample templates to train. According to this framework, the gesturing hand is detected using a motion detection method based on frame difference and depth segmentation. Trajectory recognition instead is performed using the nearest neighbor classification framework which uses the similarity measure returned by DTW. We set four samples of hand gesture, that is, wave, click, going left, and going right. However, it is unnecessary to obtain and analyze the human body data firstly and then determine the hand position. In NiTE library, a startGestureDetection function in class HandTracker can detect simple hand gestures, for example, wave, whose parameter is the hand skeleton joint. While the hand moving from right to left or from left to right, trajectory recognition is required. At such time, we use DTW algorithm to recognize the hand gesture. Most of the existing methods focus either on static signs or on dynamic gestures. As a result, being able to classify both types of hand expressivity allows for understanding the most of hand gesture.

TABLE 1: Different number recognition results.

Number	Extracted hand contour convex	Recognized hand polygon convex
0	3	0
1	2 or 3	1
2	3	2
3	3 or 4	3
4	4	4
5	5 or 6	4

TABLE 2: Different number recognition based on auxiliary point.

Number	Extracted hand contour convex	Convex points based on auxiliary point
0	3	0
1	2 or 3	1
2	3	2
3	3 or 4	3 or 4
4	4	4
5	5 or 6	5

4. Experimental Results

In this section, many tests are made to validate the proposed static hand gesture recognition method and the Kinect-based vision system for rescue system in Figure 9. The vision system of rescue system is to combine the proposed static hand gesture and dynamic hand gesture recognition method. The users' gestures captured from the Kinect sensor are compared

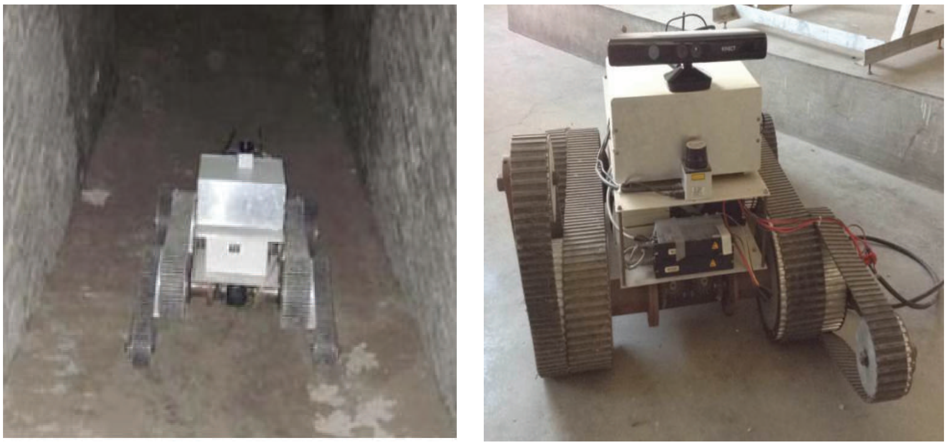


Figure 9: Kinect-based vision system.

with these gestures defined before. If the captured gesture is in the threshold range of the defined gesture, the gesture can be recognized. Then, the corresponding command is sent to control the movement direction of rescue robots through WiFi network.

Since each number presented by hand gesture has the different convex points and concave points, this paper makes use of the feature to recognize the number. The results are shown in Figure 10. Besides, the proposed method of hand polygon contour with K-curvature computing is further compared with two different recognition methods, that is, an artificial neural network and template matching. The test adopted a BP network with 11 input nets, 8 hidden nets, and 5 output nets and a template matching based on correlation coefficient. In the test, 90 depth images of hand gesture are captured at different hand angle and distance from Kinect, among which 30 images are sample data and 60 images are used to train BP network. As the second column shown in Table 3, the recognition rate of polygon contour with K-curvature computing is higher than that of the template matching method. Although both of them use less runtime, the template matching method overmuch relies on the gesture template and so has lower recognition rate. The BP network with 60 pieces of train data has higher recognition rate than the polygon contour method, as shown in the third column bracket of Table 3, while the BP network with 30 pieces of train data outside the bracket has lower value. The higher recognition rate of BP network is at the cost of a large number of samples and too much runtime, which is unfit for real-time recognition and further control. Thus, the proposed polygon contour with K-curvature computing has higher recognition rate than the BP network and template matching while being without much sacrifice of runtime. It validates that the proposed method for hand gesture recognition is feasible to implement the real-time HMI control.

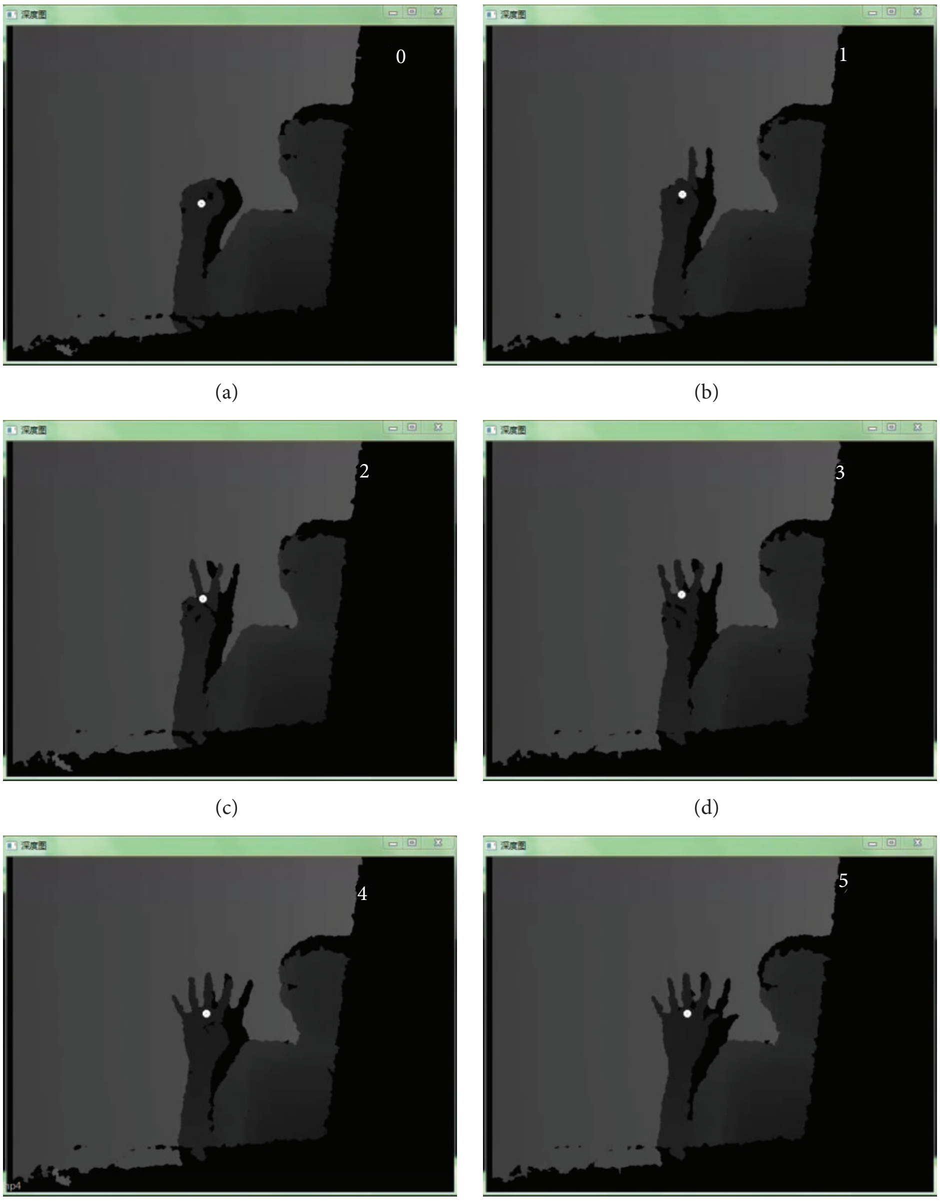
Figure 11 is an interface for HMI by which users can send a recognition result to the control system of rescue robot. The recognition of hand gesture is to mainly rely on depth information, and the RGB image provides some complementary information for hand contour extraction. Thus, as shown in Figure 11, it is achieved that even though the environment is low illuminous or darker, the recognition results of hand gesture are still better.

Table 3: Recognition rate of different recognition methods.

Number	Polygon contour recognition	BP network	Template matching
0	100%	100% (100%)	97%
1	93%	90% (100%)	83%
2	87%	77% (90%)	67%
3	77%	67% (83%)	47%
4	70%	64% (80%)	43%
5	76%	63% (77%)	40%

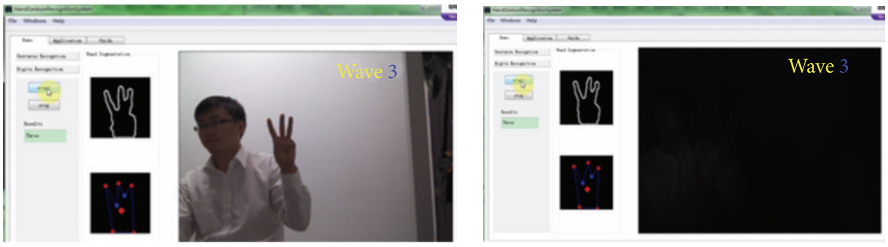
5. Conclusions

This paper applies Kinect to a control platform of rescue robot and presents Kinect-based vision system for underground mining tunnel which is low illuminous. For worse air environment, the voice recognition does not work for survivors in underground mine tunnel, because these survivors cannot take a breath. Also, for low illuminous environment, the RGB image information cannot be extracted easily. Kinect has the ability to capture the target information from its depth sensor, thereby it can work even in darkness environment. This paper proposes a static hand gesture recognition method involving hand contour extraction and K-curvature based hand polygon convex detection. Furthermore, the interface of image process combines the proposed static hand recognition and the DTW algorithm for dynamic hand recognition. Furthermore, a comparison test is made among K-curvature polygon contour method, BP neural network, and template matching, which illustrates that the proposed method has higher recognition rate while being without much sacrifice of runtime. Finally, the proposed static hand recognition and dynamic hand recognition are carried out to recognize the hand gestures about five numbers and four dynamic gestures. Experimental results validate the Kinect-based vision system.



(a) (b) (c) (d) (e) (f)

FIGURE 10: Static hand gesture recognition result.



(a) Vision system with light (b) Vision system with only PC screen light

FIGURE 11: Interface of Kinect-based vision system of rescue robot.

In conclusion, it can provide a feasible vision system of rescue robot in the low illuminous underground environment.

However, there are some constraints when applying the proposed algorithm, such as the hand position range being limited, inevitable holes, and noises around object's boundaries in the obtained images. Hence, a vision system with better performance in underground rescue robot needs to be explored more extensively. Further improved Kinect-based image process schemes or more efficient algorithms are expected to appear in future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant 51305338), the Scientific Research Plan Project of Shaanxi Province (Grant 2013JK1004), and Technology Innovation Project of Shaanxi Province (Grant 2013KTCL01-02).

References

[1] J. Suarez and R. R. Murphy, "Using the Kinect for search and rescue robotics," in *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR '12)*, College Station, Tex, USA, November 2012.

[2] D. Wei and W. Ge, "Research on one bio-inspired jumping locomotion robot for search and rescue," *International Journal of Advanced Robotic Systems*, vol. 11, no. 10, pp. 1–10, 2014.

[3] Z. Yao and Z. Ren, "Path planning for coalmine rescue robot based on hybrid adaptive artificial fish swarm algorithm," *International Journal of Control and Automation*, vol. 7, no. 8, pp. 1–12, 2014.

[4] A. Poncela and L. Gallardo-Estrella, "Command-based voice teleoperation of a mobile robot via a human-robot interface," *Robotica*, vol. 33, no. 1, pp. 1–18, 2015.

[5] W.-M. Liu and L.-H. Wang, "The soccer robot the auto-adapted threshold value method based on HSI and RGB," in *Proceedings of the 4th International Conference on Intelligent Computation Technology and Automation (ICICTA '11)*, vol. 1, pp. 283–286, March 2011.

[6] S. I. A. Shah, E. N. Johnson, A. Wu, and Y. Watanabe, "Single sensor-based 3D feature point location for a small flying robot application using one camera," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 228, no. 9, pp. 1668–1689, 2014.

[7] R. Lagisetty, N. K. Philip, R. Padhi, and M. S. Bhat, "Object detection and obstacle avoidance for mobile robot using stereo camera," in *Proceedings of the IEEE International Conference on Control Applications*, pp. 605–610, Hyderabad, India, August 2013.

[8] Y. You, T. Tang, and Y. Wang, "When arduino meets Kinect: an intelligent ambient home entertainment environment," in *Proceedings of the 6th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC '14)*, vol. 2, pp. 150–153, Hangzhou, China, August 2014.

[9] V. A. Prabhu, A. Tiwari, W. Hutabarat, and C. Turner, "Monitoring and digitising human-workpiece interactions during a manual manufacturing assembly operation using Kinect TM," *Key Engineering Materials, Advanced Design and Manufacture*, vol. 572, no. 1, pp. 609–612, 2014.

[10] Y. Yang, F. Pu, Y. Li, S. Li, Y. Fan, and D. Li, "Reliability and validity of kinect RGB-D sensor for assessing standing balance," *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1633–1638, 2014.

[11] K. Qian, H. Yang, and J. Niu, "Developing a gesture based remote human-robot interaction system using kinect," *International Journal of Smart Home*, vol. 7, no. 4, pp. 203–208, 2013.

[12] N. Abdelkrim, K. Issam, K. Lyes et al., "Fuzzy logic controllers for mobile robot navigation in unknown environment using Kinect sensor," in *Proceedings of the International Conference on Systems, Signals, and Image Processing*, pp. 75–78, 2014.

[13] B.-F. Wu, C.-L. Jen, W.-F. Li, T.-Y. Tsou, P.-Y. Tseng, and K.-T. Hsiao, "RGB-D sensor based SLAM and human tracking with Bayesian framework for wheelchair robots," in *Proceedings of the International Conference on Advanced Robotics and Intelligent Systems (ARIS '13)*, pp. 110–115, IEEE, Tainan, Taiwan, May-June 2013.

[14] G. Du and P. Zhang, "Markerless human-robot interface for dual robot manipulators using Kinect sensor," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 2, pp. 150–159, 2014.

[15] K. Buckstone, L. Judd, N. Orlowski, M. Tayler-Grint, R. Williams, and E. Zauls, "Warwick mobile robotics: urban search and rescue robot," Tech. Rep., University of Warwick, Coventry, UK, 2012.

[16] J. L. Raheja, A. Chaudhary, and K. Singal, "Tracking of fingertips and centers of palm using kinect," in *Proceedings of the 3rd International Conference on Computational Intelligence, Modelling and Simulation (CIMSim '11)*, pp. 248–252, September 2011.

[17] T. T. Thanh, F. Chen, K. Kotani, and H.-B. Le, "Extraction of discriminative patterns from skeleton sequences for human action recognition," in *Proceedings of the IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF '12)*, pp. 1–6, Ho Chi Minh City, Vietnam, February-March 2012.

[18] P. Doliotis, V. Athitsos, D. I. Kosmopoulos, and S. Perantonis, "Hand shape and 3D pose estimation using depth data from a single cluttered frame," in *Proceedings of the International Symposium on Visual Computing (ISVC '12), Crete, Greece, July 2012*, vol. 7431 of *Lecture Notes in Computer Science*, pp. 148–158, Springer, Berlin, Germany, 2012.

[19] T. Wan, Y. Wang, and J. Li, "Hand gesture recognition system using depth data," in *Proceedings of the 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet '12)*, pp. 1063–1066, Yichang, China, April 2012.

[20] F. Pedersoli, S. Benini, N. Adami, and R. Leonardi, "XKin: an open source framework for hand pose and gesture recognition using kinect," *The Visual Computer*, vol. 30, no. 10, pp. 1107–1122, 2014.

[21] N. Chahal and S. Chaudhury, "High quality depth map estimation by kinect upsampling and hole filling using RGB features and mutual information," in *Proceedings of the 4th National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG '13)*, pp. 1–4, December 2013.