3D Pointing Gesture Recognition for Human-Robot Interaction

Yuhui Lai[1], Chen Wang[2], Yanan Li[3], Shuzhi Sam Ge[2], Deqing Huang[4]

1. University of Illinois Urbana Champaign, Champaign IL, 61820
E-mail: lai35@illinois.edu

2. Department of Electrical & Computer Engineering, National University of Singapore, Singapore, 117576
E-mail: {wang_chen09,samge}@nus.edu.sg

3. Department of Bioengineering, Imperial College London, London, SW7 2AZ
E-mail: hit.li.yn@gmail.com

4. School of Electrical Engineering, Southwest Jiaotong University, Chengdu, 610031
E-mail: elehd2012@gmail.com

Abstract: In this paper, a pointing gesture recognition method is proposed for human-robot interaction. The pointing direction of the human partner is obtained by extracting the joint coordinates and computing through vector calculations. 3D to 2D mapping is implemented to build a top-view 2D map with respect to the actual ground circumstance. Using this method, robot is able to interpret the human partner's 3D pointing gesture based on the coordinate information of his/her shoulder and hand. Besides this, speed control of robot can be achieved by adjusting the position of the human partner's hand relative to the head. The recognition performance and viability of the system are tested through quantitative experiments.

Key Words: Gesture Recognition, Human-Robot Interaction, Robot Control

1 INTRODUCTION

Body postures are powerful means for humans to convey information. They allow individuals to communicate a variety of feelings and thoughts, from contempt and hostility to approval and affection [1, 2]. Different from verbal communications, body gestures employ body poses to express intentions. One essential way of displaying spatial knowledge is by pointing, utilizing arm and hand to indicate a specific direction. In [3], it has been shown that even people in different societies and speaking different languages used cardinal directions to describe things and ended up with vectors in appropriate divisions of the horizontal plane. In terms of robotics, due to the fact that pointing gesture is a universal language to indicate directions, there is no barrier for people in different backgrounds interacting with robots using pointing gestures [4]. Earlier studies have already shown that pointing gesture is particularly useful for object localization [5] and direction specification [6], and thus provides intuitive information which can be used in human-robot interaction [7].

Many research works in the literature have focused on the pointing gesture recognition [8] and its applications beyond robotics [9, 10]. In terms of capturing human postures, three types of vision systems have been widely used, namely, Time of Flight (ToF) camera, stereo camera, and Kinect [11]. Researchers have also built various computational models for pointing gesture detection, such as active appearance model [12], cascade Hidden Markov Models (HMMs) [13], and parametric HMMs [14]. In order to extract pointing vectors from human body, head-hand line [15], wrist-hand line, and line of sight [16] were employed. A real-time visual system was proposed in [15], where color and disparity information was retrieved from a stereo camera and used to locate hands and head. In [17], a single camera was used in recognizing pointing gestures without markers. The user's pointing direction toward the cell of a grid on a screen with his/her hand being fully stretched was successfully identified. Although this system successfully built up a natural interface for human-computer interaction, the setup and calibration of a single camera were time-consuming and tedious. Unlike above methods, a different way in analyzing the pointing direction was introduced in [18]. They used a ToF camera to detect 3D point clouds, based on which they were able to figure out the location of the head, segment the body, and localize the elbow, hand, and shoulder. However, it is known that ToF cameras suffer from two major problems: a low resolution and a low sensitivity, which result in a high-level noise [19]. Additionally, background lights may cause problems in outdoor environments.

This paper aims to develop an efficient and economical real-time 3D pointing gesture recognition system for human-robot interaction. We propose a method that only uses hand and shoulder coordinates retrieved from Kinect. 3D to 2D mapping and pointing vector identification are realized. To cope with the problem by random arm lifting, we regard the first lifted arm as the pointing arm and others for speed control of robot. Compared to [15], our system demonstrates a better accuracy in direction detection and spares the trouble of stereo camera calibration and setups. Instead of using HMMs to predict pointing gesture, which

This work is submitted to the invited session entitled "Learning and Control in Human-Robot Interaction".

is computationally intractable, we adopt an approach which measures the vertical distance between the lifted hand and the corresponding shoulder. This method results in a simple way of detecting the pointing gesture and ease of computational cost due to massive training data.
Based on the above discussions, we highlight the contributions of this paper as follows:

(i) a real-time 3D pointing gesture recognition method is developed, which is considerably efficient in detecting pointing gestures as it eases the pressure from the training procedure yet it still achieves a high recognition accuracy;

(ii) speed control of robot is achieved by calculating the relative vertical position of the human partner's hand to his/her head; and

(iii) an interface is built up to enable the efficient task execution through human-robot interaction.

The rest of this paper is structured as follows. In Section 2, the system setup is briefly introduced and the skeletal tracking function of Kinect is described. In Section 3, the proposed method of 3D pointing gesture recognition is elaborated. In Section 4, experiments in a human-robot interaction scenario are implemented to verify the accuracy and viability of the proposed method. Section 5 concludes the paper.

2 SYSTEM OVERVIEW

The proposed method of pointing gesture recognition consists of three major steps: skeletal joints detection, 3D to 2D mapping, and pointing vector extraction. Once we obtain the joints data from Kinect, the 3D coordinates of joints can be used to represent the pointing vector. However, in reality we are more concerned of the direction that human is pointing at in the horizontal plane. Therefore, a top-view 2D map will be constructed in order to obtain the pointing vector. The shoulder-hand line in the 2D map will be assigned as a vector to recognize the pointing direction and navigate the robot. A speed control which takes advantage of the relative height of the other hand to the head is also introduced in this framework. Figure 1 depicts a typical human-robot interaction scenario utilizing the proposed framework, where a human partner stands in front of a Kinect and uses his pointing gesture to navigate the robot.

Figure 1: A human partner interacts with a robot using Kinect.

2.1 Kinect

Kinect is a state-of-the-art motion sensing device by Microsoft with a low cost [11]. The device features a RGB camera, a depth sensor, and a multi-array microphone running a proprietary software, which provides functions such as full-body 3D motion capture, facial recognition, and voice recognition. With the ability to capture 3D motions, Kinect can track human motions and capture up to 24 human skeletal joints data in 3D, which provides users with intuitive information about human body gestures. Microsoft provides a standard re-calibration procedure to ensure that the sensor is able to track the body correctly and robustly. Due to the fact that it is relatively cheap, compatible, and mobile, and performs even better in capturing motions compared to stereo cameras or ToF cameras, it becomes the top choice in building a pointing gesture recognition system in this paper.

2.2 Skeletal Joints Capture

Typical point clouds of a human body captured by Kinect are shown in Figure 2. Given the coordinate information of hand and shoulder of the human partner, the pointing vector can be calculated while the coordinates of the head and the other hand can be used as important parameters in speed control of robot. In order to improve the robustness of the proposed framework, only the first scanned skeleton is used for pointing gesture recognition.

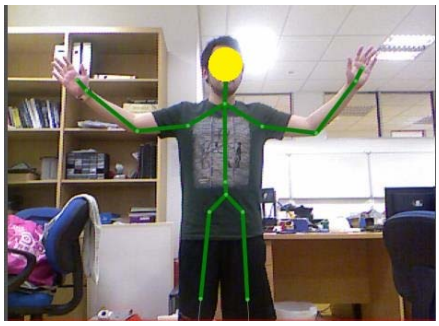

Figure 2: Typical point clouds of a human body captured by Kinect.

3 POINTING GESTURE RECOGNITION

In order to interpret the pointing gesture, we divide the recognition procedure into following steps: i) initialization of the pointing gesture detector whenever it recognizes a pointing gesture; and ii) approximation of the pointing direction using the extracted pointing vector.

3.1 Pointing Gesture Detection

There are extensive means for a narrator to exploit in indicating a direction, and the performance of a pointing gesture may vary and be limited to physical surroundings. Skilled narrators may even use self-created hand movements in explaining directions to interlocutors, so the seemingly unproblematic notion of directions becomes sophisticated [13]. In this section, we aim to unravel this complexity in notion of directions and variety of performance

in terms of pointing gesture. In order to distinguish a pointing gesture from other unintentional hand movements, we comprehend certain requirements that a hand movement has to meet to be recognized as a pointing gesture. Without these requirements, background noise or other random hand motions may interfere the identification process and reduce the effectiveness in human-robot interaction.

People are not prone to randomly lift their hands to the same level as shoulders for a while, as this motion is likely to cause discomforts. We utilize this finding to set up our triggering mode of detection initialization. Also, to avoid complexity of notion and construct a uniform standard, we propose the following criterion: a meaningful pointing gesture will be recognized only when a human partner lifts either his/her left hand or right hand to the same level as his/her shoulder. This will lead to a relatively computational method. In order to comprise flexibility to this method, we allow for a range of relative vertical distance between the vertical position of the hand and shoulder, e.g., 13cm. The holding phase is also required to initiate the detection process which must last for more than a certain time, e.g., 1s.

In order to distinguish the arm used for direction pointing and the arm for speed adjustment, we compare the lengths of both arms in a top-view 2D map (as shown in Figure 3). In a real-world situation, due to limited physical conditions or special circumstances, determining the pointing arm will be complicated. In the proposed method, we use a simple strategy that as long as the pointing arm is stretching further than the other arm on the 2D map, then the pointing arm is identified. This strategy is effective in most cases, even when the human partner does not fully stretch his/her one arm to indicate a direction and the other arm is under a slight random motion.

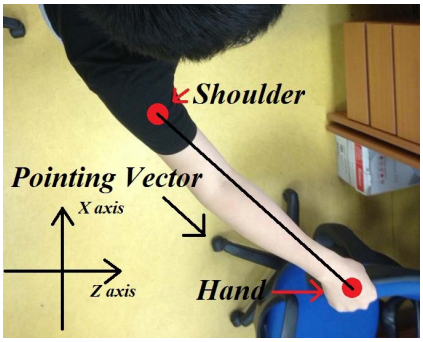

Figure 3: Top-view 2D mapping.

3.2 Pointing Direction Recognition

In the literature, line of sight or head-hand line are widely used for pointing direction recognition [15, 13, 16]. However, these methods may fail in some situations where people cannot raise their arms in the same level as eyes or fully stretch their arms. To cope with this issue, only shoulder and hand coordinates are used in the proposed method to interpret the pointing vector. In particular, we extract the horizontal vector connecting the hand and shoulder regardless of their vertical positions.

First, we build a top-view 2D X-Z coordinate map, then extract X and Z coordinates of the hand and shoulder to initiate the vector calculation. As shown in Figure 3, a shoulder-hand line has been retrieved from the point clouds and a unit vector in the direction of shoulder-hand line is calculated using the following formulas.

Denote the X and Z coordinates of the hand as X_{hs} and Z_{hs}, respectively, and the X and Z coordinates of the shoulder as X_s and Z_s, respectively. Then, we have

$$\begin{aligned} \overrightarrow{X} &= X_{hs} - X_s \\ \overrightarrow{Z} &= Z_{hs} - Z_s \\ \overrightarrow{P_x} &= \overrightarrow{X} / \sqrt{(\|\overrightarrow{X}\|^2 + \|\overrightarrow{Z}\|^2)} \\ \overrightarrow{P_z} &= \overrightarrow{Z} / \sqrt{(\|\overrightarrow{X}\|^2 + \|\overrightarrow{Z}\|^2)} \end{aligned} \quad (1)$$

where $\overrightarrow{X}$ and $\overrightarrow{Z}$ represent the pointing vectors in X and Z directions, respectively, and $\overrightarrow{P_x}$ and $\overrightarrow{P_z}$ are the horizontal and vertical components in X and Z directions of the unit pointing vector $\overrightarrow{P}$, respectively. The human partner's pointing direction is thus obtained by calculating $\overrightarrow{P}$.

3.3 Speed Control

Meanwhile, the other hand which has not been used for direction pointing can be used for speed control of robot. In particular, if a human partner's right arm is pointing to some direction, then his/her left hand's relative vertical position to his/her head can be used to determine the speed of the robot. According to the skeletal data, the vertical distance from the human partner's hand to his/her head normally will fall into a range, e.g., from -0.6m to 0.6m. Define two speeds: V_{min} (pixels/second) and V_{max} (pixels/second), corresponding to two distances of -0.6m and 0.6m, respectively. Then, we can modulate the speed of robot using a linear equation. Denote Y_{hs} and Y_h as the Y coordinates of the other hand and the head, respectively. Then, we have

$$\begin{aligned} H &= \|Y_{hs} - Y_h\| \\ D &= \frac{V_{max} - V_{min}}{1.2}(H + 0.6) + V_{min} \end{aligned} \quad (2)$$

where H is the vertical distance between the human partner's head and the other hand, and D is the resulted speed of robot.

4 EXPERIMENTS

In this section, two sets of experiments are conducted to test the accuracy of the proposed method in pointing gesture recognition and the viability of the developed system in robot navigation.

4.1 Pointing Gesture Recognition

In order to test the accuracy in estimating the pointing direction in an indoor environment, we measured the error angles between the assigned directions and the actual results in real situations. Kinect was placed on a desk such that the sensors were exactly towards south and the whole body of human partner could be scanned. The human partner was asked to stand in a distance of 1m in front of Kinect

and align his body towards it, as illustrated in Figure 4. In order to meet the system requirement for detection of pointing gesture, the human partner raised his/her hand to the same level as his/her shoulder, as shown in Figures 5 and 6. We tested on 7 ideal pointing directions to verify the effectiveness of the method, i.e., north, northeast, east, southeast, southwest, west, and northwest, as also shown in Figures 5 and 6. The reason that we did not test the south direction is that Kinect cannot detect the pointing hand when the human partner stands towards it and stretches his/her arm behind the body.

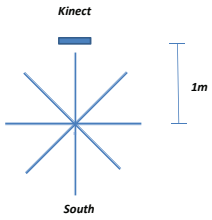

Figure 4: Human partner stood at the origin of the cartesian coordinates map.

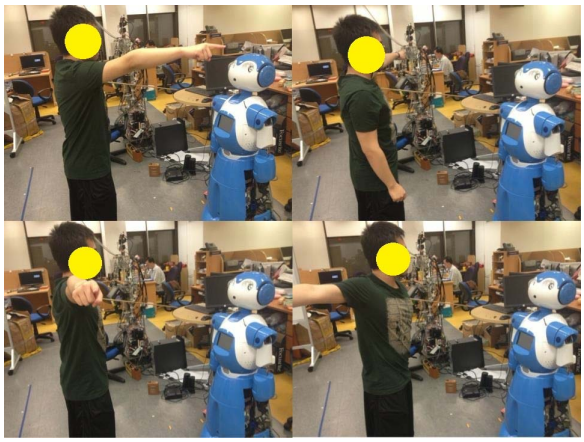

Figure 5: In a typical human-robot interaction scenario, a human partner performed pointing gesture in directions of northeast, northwest, east, and southeast in front of a robot.

10 trials were carried out for each ideal pointing direction in this experiment. The experiment results of error angles are summarized in Table 1. From Table 1, we can see that the shoulder-hand line approximation yields an overall average error angle of 3.59 degrees and a standard deviation of 2.18 degrees, which are acceptable in a human-robot interaction scenario. The box plot for Table 1 is shown in Figure 7. It is found that the median for the error angle in the direction of northwest is 4.67 degrees and of southwest is 5.61 degrees, which are higher compared to the medians of other directions. There are also variations of larger amounts in these two directions. These results are due to the fact that we did not calibrate the system after each trial. Results of time used for detecting pointing gestures for each trial are summarized in Table 2. The average time for detecting and calculating the pointing gesture for 70 trials is 0.39 seconds, which is dependent on the performance of

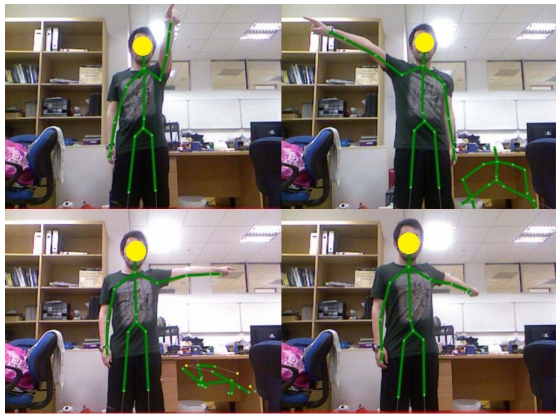

Figure 6: Skeletal tracking images corresponding to the above human-robot interaction scenario extracted from Kinect.

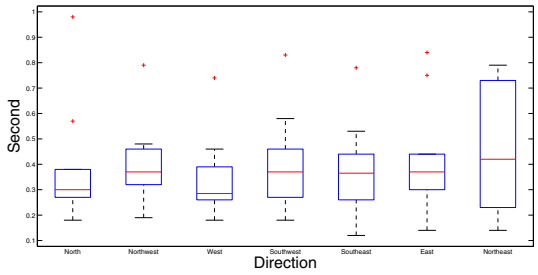

Figure 7: Box plot for Table 1.

Kinect itself. Compared to an average error angle of 25 degrees in [15] where the head-hand line approximation was adopted, we achieved a higher accuracy in approximating the pointing direction. As shown in Figure 8, there is a variation of a small amount within a range of 0 to 0.13 seconds in terms of the medians in calculating the pointing vector in all 7 directions. In terms of gesture detection time, individual pointing gesture difference would also affect the performance of our system. However, the error range is fairly small and acceptable. There are also situations in which it took almost 0.9 seconds, which is because Kinect is not capable of reading the shoulder coordinates if the shoulder-hand line is directly orthogonal to it.

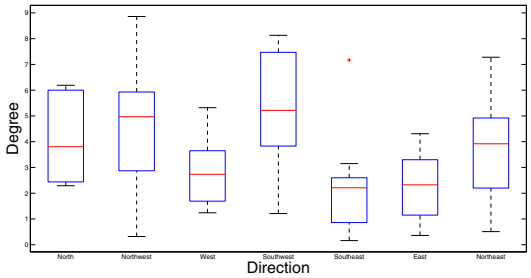

Figure 8: Box plot for Table 2.

The above results demonstrate the accuracy of the proposed method in detecting direction in a 2D space. However, no-

	North	Northwest	West	Southwest	Southeast	East	Northeast
Trial 1	2.41	5.52	1.24	5.61	0.16	3.30	4.14
Trial 2	2.44	8.86	2.38	7.47	2.60	1.15	4.19
Trial 3	2.29	5.27	3.09	5.82	0.86	3.14	4.92
Trial 4	6.14	8.34	1.69	7.96	2.52	0.36	0.51
Trial 5	6.00	0.32	1.25	3.83	2.44	2.32	3.70
Trial 6	2.79	2.87	3.27	4.82	1.78	3.37	2.13
Trial 7	3.46	4.67	4.25	1.21	7.17	4.31	5.23
Trial 8	5.16	0.38	5.32	8.13	3.15	1.42	7.28
Trial 9	4.15	2.94	2.10	3.83	1.98	0.39	3.35
Trial 10	6.19	5.93	3.65	4.12	0.42	2.33	2.20
Average	4.10	4.69	2.82	5.28	2.31	2.21	3.77

Table 1: Angle error between the ideal assigned direction and the actual performance in a real situation (all units in degrees).

	North	Northwest	West	Southwest	Southeast	East	Northeast
Trial 1	0.21	0.46	0.28	0.37	0.78	0.30	0.79
Trial 2	0.34	0.79	0.46	0.25	0.27	0.15	0.23
Trial 3	0.31	0.38	0.18	0.46	0.53	0.14	0.14
Trial 4	0.57	0.41	0.39	0.27	0.37	0.36	0.78
Trial 5	0.28	0.36	0.27	0.83	0.36	0.32	0.41
Trial 6	0.18	0.32	0.74	0.18	0.44	0.38	0.17
Trial 7	0.29	0.33	0.18	0.29	0.21	0.39	0.39
Trial 8	0.38	0.48	0.29	0.40	0.39	0.44	0.57
Trial 9	0.98	0.19	0.38	0.37	0.26	0.84	0.43
Trial 10	0.27	0.22	0.26	0.58	0.12	0.75	0.73
Average	0.38	0.39	0.34	0.39	0.37	0.41	0.46

Table 2: Time used for detecting pointing gestures for each trial corresponding to Table 1 (all units in seconds).

ticeable enhancement is needed in the sense that the recognition system has placed an impose on a standard pointing gesture performing which is unfriendly to the first time user. Also, there are a few disadvantages of using Kinect. First, Kinect cannot capture the coordinates of the hand and shoulder when the human partner stretches his/her arm behind the body, as discussed above. Second, the human partner cannot wear reflective clothes, due to the fact that Kinect projects the infrared (IR) laser and calculates the depth using IR pattern, which is invisible on transparent or reflective surfaces. Third, in the bottom left side of the Figure 6, there is a random skeleton appearing due to the existence of noise. It may cause problems for Kinect to detect the desired skeleton if people pass by the human partner.

4.2 Robot Navigation

We conducted anther experiment to test on the viability of the developed system in a virtual human-robot interaction scenario, which is shown in Figure 9. In this scenario, a robot travelled in a hotel hallway without any prior map knowledge, starting at left bottom corner and ending at left top corner. Information of assigning directions was given by the human partner at each corner when direction change was necessary, and help was offered whenever the robot hit the wall. The visual information was received under uniform lighting condition from Kinect which was on top of the robot. In order to evaluate the robustness of this system, we conducted this experiment using the same map for three times with three human partners. These human partners did not have any knowledge of the system setup and were told to give direction information to the robot by performing pointing gestures at the shoulder level. Additionally, human partners were able to adjust the speed of the robot by raising their non-pointing hands.

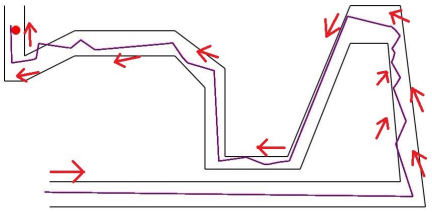

Figure 9: Test result for human partner 1 who collaborated with a robot in a hotel hallway in a virtual scenario. The red arrow (not all direction instructions are shown in the figure) indicates the human partner performing pointing direction to navigate the robot, the purple line illustrates the path that the robot has actually taken under guidance, and the red eclipse denotes the position of the robot.

The completion time, average speed and number of guidance instances are summarized in Table 3. Ideally 10 direction instructions were expected as there were ten corners. However, it ended up with an average of 18.67 pointing gestures in assisting the robot navigating through the hallway. The average completion time by three different human partners is 75 seconds. During the experiment, the human partner was not always able to point to the direction that is parallel to the path. Besides, even a small angle error made in the recognition phase would lead the robot to hit the wall and thus result in extra help. In this sense,

	Completion time (second)	Number of pointing gestures performed	Average speed of each trial (pixels/second)
Human partner 1	73	18	2.56
Human partner 2	87	22	2.15
Human partner 3	65	16	2.88
Average	75	18.67	2.53

Table 3: Test results for three human partners interacting with robot in the same scenario using pointing gestures in order to guide it traveling through the hallway.

it is necessary to enhance the performance of the system by incorporating a feedback control. The total distance in this experiment is 187 pixels, and the average speed for three trials is 2.53 pixels/second. The speed control can be scaled based on the actual need in practice. From the above experiment results, we can conclude that the virtual robot eventually arrived at the end point of the hallway through the human partner's guidance with pointing gestures. Although there were differences between each individual human partner in performing pointing gestures, the robot was still capable of recognizing the pointing direction and completed the task. However, the lighting condition may place a constraint on the recognition rate since the IR sensor of Kinect performs relatively poorer in obtaining the 3D coordinates of human joints outdoors.

5 CONCLUSIONS

In this paper, we presented a 3D pointing gesture recognition system, which enabled a robot to understand 3D human pointing gestures in real time. We have verified its validity in a designed scenario where a virtual robot successfully travelled through the hotel hallway by interpreting the human partner's pointing gestures. We have also achieved an acceptable average error angle between the direction interpreted by the robot and the expected direction. Additionally, a speed control was introduced to improve the system performance. This system has endowed us an intuitive way to interact with a robot. Future applications of this system can be explored in different areas, e.g., guiding a vacuum to clean certain areas, or navigating a trolley in a grocery store.

REFERENCES

[1] Y. Chuang, L. Chen, and G. Chen, "Saliency-guided improvement for hand posture detection and recognition," *Neurocomputing*, 2014.

[2] D. Kelly, J. McDonald, and C. Markham, "A person independent system for recognition of hand postures used in sign language," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1359–1368, 2010.

[3] C. Rich, B. Ponsler, A. Holroyd, and C. L. Sidner, "Recognizing engagement in human-robot interaction," in *the 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2010*, pp. 375–382, IEEE, 2010.

[4] Y. Li, K. Tee, S. S. Ge, and H. Li, "Building companionship through human-robot collaboration," in *Social Robotics* (G. Herrmann, M. Pearson, A. Lenz, P. Bremner, A. Spiers, and U. Leonards, eds.), vol. 8239 of *Lecture Notes in Computer Science*, pp. 1–7, Springer International Publishing, 2013.

[5] R. El-laithy, J. Huang, and M. Yeh, "Study on the use of microsoft kinect for robotics applications," in *Proceedings of the Position Location and Navigation Symposium*, pp. 1280–1288, 2012.

[6] C.-Y. Chien, C.-L. Huang, and C.-M. Fu, "A vision-based real-time pointing arm gesture tracking and recognition system," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 983–986, 2007.

[7] E. Sato, T. Yamaguchi, and F. Harashima, "Natural interface using pointing behavior for human-robot gestural interaction," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 2, pp. 1105–1112, 2007.

[8] T. Kirishima, K. Sato, and K. Chihara, "Real-time gesture recognition by learning and selective control of visual interest points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 351–364, 2005.

[9] Z. Černeková, C. Malerczyk, N. Nikolaidis, and I. Pitas, "Single camera pointing gesture recognition for interaction in edutainment applications," in *Proceedings of the Spring Conference on Computer Graphics*, pp. 121–125, 2008.

[10] D. Geer, "Will gesture recognition technology point the way?," *Computer*, vol. 37, no. 10, pp. 20–23, 2004.

[11] Microsoft, "http://www.microsoft.com/en-us/kinectforwindows/," 2013.

[12] M. Reale, S. Canavan, L. Yin, K. Hu, and T. Hung, "A multi-gesture interaction system using a 3-D iris disk model for gaze estimation and an active appearance model for 3D hand pointing," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 474–486, 2011.

[13] C.-B. Park, M.-C. Roh, and S.-W. Lee, "Real-time 3D pointing gesture recognition in mobile space," in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1–6, 2008.

[14] A. D. Wilson and A. F. Bobick, "Parametric hidden markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.

[15] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human–robot interaction," *Image and Vision Computing*, vol. 25, no. 12, pp. 1875–1884, 2007.

[16] R. Kehl and L. Van Gool, "Real-time pointing gesture recognition for an immersive environment," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 577–582, 2004.

[17] Z. Cernekova, N. Nikolaidis, and I. Pitas, "Single camera pointing gesture recognition using spatial features and support vector machines," in *Proceedings of the 15th European Signal Processing Conference*, pp. 130–134, 2007.

[18] D. Droeschel, J. Stuckler, and S. Behnke, "Learning to interpret pointing gestures with a time-of-flight camera," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, pp. 481–488, 2011.

[19] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "Lidarboost: Depth superresolution for ToF 3D shape scanning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 343–350, 2009.