

# **Control Robots using Red Hands: A Human-Robot Interaction System using Human Hand Motions**

**Mostafa Korashy**

Department of Information Technology,  
Assiut University, Egypt

**Mahmoud Afifi**

Department of Information Technology,  
Assiut University, Egypt

## **ABSTRACT**

Human-robot interaction is an evolving area of research in the past few years. Human-robot interaction deals with how humans can interact with, send data to, or receive data from robots. One of the major obstacles in this field is how the robot can obtain the depth information of the surrounding objects. Few years ago, Microsoft has released a depth sensor that computes the depth information using IR rays. Many researches are conducted to control robots using depth sensors, such as Microsoft Kinect and Asus Xtion. Although depth sensors are considered to be low cost, it may be unavailable for many users. In this work, we develop a low-cost system for controlling robots (iRobot) with a web-cam and just red markers on the user's hands. Our system requires no extra devices or hardware or other complex technologies. Experimental results of the proposed system demonstrate good results compared to those provided by depth sensors.

## **General Terms**

Human-robot Interaction, Robotics, Computer Vision.

## **Keywords**

HRI, iRobot Create, Color detection.

## **1. INTRODUCTION**

Fully-automated robots require gathering information about their environment for making decisions and taking correct actions. In another direction, semi-automated mobile robots require support from humans. This support is represented by controlling the robots via humans for aiding them to take correct actions. Starting in the 1980s, the problem of controlling mobile robots has paid attention to be used in many applications[1]. Nowadays, human-robot interaction is moving from using mouse or keyboard to be sensitive to the real environment of the robot [2]. Speech recognition was used for controlling robots using voice commands, such as J. N. Pires [3]. Improving the speech recognition by using computer vision techniques was presented by J.B.Gomez et al. [4] for controlling a robot using mouth gestures and voice commands. Other methods present human-robot interaction using full human movements, such as L. Zheng et al. [5] who used Hidden Markov Model (HMM) for classifying each motion to control the robot using the user's movements. In another way of improvement, some methods focus on hand gestures recognition only for controlling robots, such as M.C. Kao et al. [6] who presented an interaction system for controlling the humanoid robot using hand and arm movements. Electromyography (EMG) technique is used by P. Shenoy et al. [7] who used EMG signals for controlling a robotic arm by the real movements of the user's arm. Few

years ago, Microsoft has released a depth sensor that computes the depth information using IR rays [8]. Many researches are conducted to improve robot's sensibility and control issues using depth sensors. Depth sensors are used to gather depth information about the robot's environment for many applications, such as object recognition, collision detection, and gestures control [9]. In gestures control, depth sensors are used for recognizing human gestures and translating those gestures into specific actions that are performed by robots, such as Q. Li et al.[10] who used Microsoft Kinect depth sensor for controlling a robot using recognized human gestures. G. Du et al. [11] used Microsoft Kinect depth sensor for tracking motions of the user's hand and controlling a picker robot which mimics the user's hand gestures for picking an object. Despite the good performance of depth sensors, it maybe unavailable for many users. In another direction, computer vision techniques can achieve good hand gestures recognition such as W. Chen [12] and H. Yeo et al. [13]who presented a hand gesture estimation using a web-cam with some assumptions and limitations, such as in [12] the contour of hand and forearm must be the only contour appears to the web-cam, and in [13] inadvertent rotations of the hand along Y axis lead to some problems in the recognition process. In addition, lighting conditions are taken into account as a serious factor that has effects on the accuracy of the recognition process [13]. Their techniques also based on skin detection, so there is a restriction on the amount of skin visible from the body of the user. In spite of their real-time recognition of hand's movements and gestures, their techniques give inaccurate results in robot's movement because of frequently shakes that occur from imponderable centroid of the palm.

In this work, we present a human-robot interaction system using human hand motions. The proposed system is a low-cost human-robot interaction system that requires only two web-cams and two red markers. The two red markers are held by the user's hands. Color and object detection, segmentation, morphological operations and some activity recognition basics are performed for estimating the movement of the user's hands to control robot's motions. The proposed system requires no depth sensors or extra devices for controlling the movement of the robot. Experimental results of the proposed system demonstrate good results compared to those provided by using depth sensors.

The remainder of this paper is organized as follows: In section 2, we present the details of the proposed system. In section 3, we present the experimental results of the proposed system, comparisons, and the applications of the proposed system. Finally, the paper is concluded in section 4.

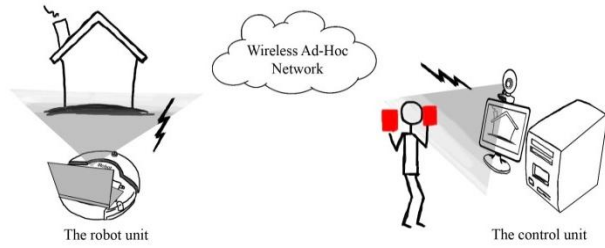


Fig 1: The overview of the proposed system.

## 2. METHODOLOGY

In this section, the methodology of the proposed system is presented. The hardware components of the proposed system are two web-cams, one of them is connected to a PC (control unit) and another one is connected to a laptop that is located above a robot (robot unit), and two red markers. The reason of using the red markers is to flee away from the complexity of the hand's shape for getting robust centroid without any accidental shaking. Fig. 1 shows the overview of the proposed system. The poses of the user's hands are captured by the control unit using the web-cam. Red hands detection is applied to the received frames. After that, segmentation and morphological operations are applied for determining the centroid of each hand. From the extracted centroid, the control unit determines the required command for sending it to the robot unit via a wireless Ad-Hoc network. Next, the robot unit transfers the command to the robot. The user in the control unit side sees what the robot sees through a web-cam that is connected to the laptop above the robot. In the next subsections, the details of the control unit and robot unit are presented.

### 2.1 The Control Unit

In control unit, three main tasks are performed:

#### 2.1.1 Object Detection

In this work, a red marker of rectangular shape is detected using two stages. First we detect all the rectangular shapes. The rectangular shapes are then filtered to only red rectangles. An image of two objects corresponding to the two markers is generated after the previous stages. Connected component labeling algorithms is applied to the resulted image to detect each marker individually.

#### 2.1.2 Centroid Computation

After completing the object (marker) detection, the centroid of each object is computed. Assuming that the resulting image is  $IMG(i,j)$ , then the centroid pixel of each object ( $I_C, J_C$ ) is computed using the following equations [14]

$$I_C = \frac{\sum_i \sum_j I \times IMG(i,j)}{\sum_i \sum_j IMG(i,j)} \quad (1)$$

$$J_C = \frac{\sum_i \sum_j J \times IMG(i,j)}{\sum_i \sum_j IMG(i,j)} \quad (2)$$

Because M Choras' et al. [14] assumed that the equation is applied to binary image, we modify Equ. 1 and Equ. 2 for operating with colored images, so they became:

$$I_C = \frac{\sum_i \sum_j I \times IMG_R(i,j) / 255}{\sum_i \sum_j IMG_R(i,j) / 255} \quad (3)$$



Fig 2: The predefined signs from the user for controlling the robot.

$$J_C = \frac{\sum_i \sum_j J \times IMG_R(i,j) / 255}{\sum_i \sum_j IMG_R(i,j) / 255} \quad (4)$$

#### 2.1.3 Command Recognition

After detecting the center of each object, we compute the absolute difference between the y-coordinate ( $y_1, y_2$ ) of each center. Then we compare the positions of the centers and commands are recognized according to the following conditions:

$$|y_2 - y_1| = \begin{cases} > T \text{ AND } (y_1 < y_2) \text{ Move right} \\ > T \text{ AND } (y_1 > y_2) \text{ Move left} \\ < T \text{ Move forward} \end{cases} \quad (5)$$

Where  $T$  is a predefined threshold that is calculated experimentally. Algo. 1 describes the steps that performed in the control unit.

### 2.2 The Robot Unit

This unit is mainly responsible for interfacing with the robot. The control unit's commands are sent to the robot through this unit. The unit also sends a video stream to the control unit to enable the user to view the remote environment in which the robot move. In Algo. 2, steps that are performed in the robot unit are presented.

#### 2.2.1 Connectivity

In order for the control unit to send commands and receive video stream from the robot unit, some kind of connectivity should be deployed and configured properly. Because mobile robots are used in the system, no wiring network fits our need. Wireless ad hoc network is a cheap, infrastructure less, covers a good range and easy for setup and configure. So it is appropriate for the system's purpose.

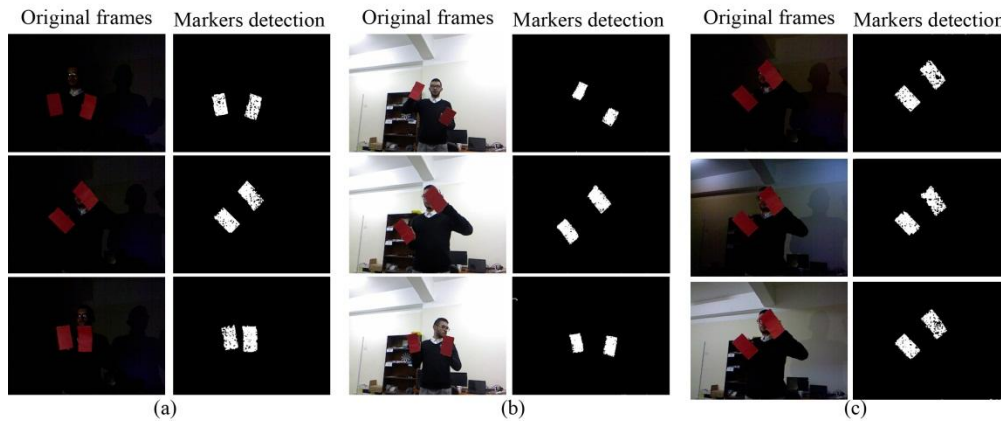


Fig 3: The results of markers detection process under different lighting conditions, such as under (a) poor light, (b) strong light, and (c) dynamic light.

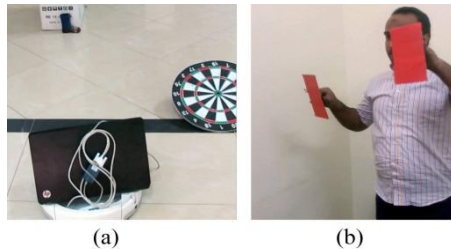


Fig 4: The results of the proposed system. (a) The control unit. (b) The robot unit.

---

**Algorithm 1** The control unit

---

- 1: Make initial System configurations (Com port, Network configuration, IPAddress,)
  - 2: Adapt the lighting conditions to detect the red colors
  - 3: Run a thread that receive the video stream from the robot unit and display it
  - 4: **while**FrameCaptured == true**do**
  - 5: Detect all rectangular shapes in the scene
  - 6: Filter the rectangles according to their colors
  - 7: Find the centroid of the detected markers
  - 8: Compare the position of the centroid of each object and recognize the command
  - 9: Send the command to the robot
  - 10: **endwhile**
- 

### 3. EXPERIMENTAL RESULTS

In this section, the experimental results, comparisons, and applications of the proposed system are presented. The proposed system is implemented using Emgu CV. The comparison with Microsoft Kinect depth sensor is implemented using Kinect for Windows SDK. The proposed

---

**Algorithm 2** The robot unit

---

- 1: Make initial System configurations (Com port, Network configuration, IPAddress,)
  - 2: Run a thread to send the video stream from the robot unit to the control unit
  - 3: Wait for commands from the control unit
  - 4: Send the command through the serial port to the robot
  - 5: go to step3
- 

system was tested on a variety of live stream videos under different lighting conditions. The experiments were done on an Intel core(TM) i7-3770 CPU @ 3.40GHz with 8 GB RAM machine at the control unit, and an Intel core(TM) i5-3317U CPU @ 1.70GHz with 4 GB RAM machine at the robot unit. The used robot is iRobot Create (<http://www.irobot.com/>). The control unit's web-cam is QuickCam@Orbit AF (<http://www.logitech.com/>)cam is true Vision HP web-cam that is embedded in the laptop.

The robot's identified commands are stop, forward, arc-forward-left, and arc-forward-right. Each command has a correspondence predefined sign by the user as shown in Fig. 2. We tested the proposed system under strong light, poor light, and dynamic light. Fig. 3 shows the results of markers detection process under different lighting conditions, where the proposed system adapts the red marker detection process based on the current light condition for finding the red markers as much as possible. Fig. 4 shows the results of the proposed system, where the robot responds to the user's hand motions.

#### 3.1 Comparisons

We used red markers for getting a robust centroid for each hand without restrictions on the visible area of the user's skin.

**Table 1. Results of comparing the proposed system using Microsoft Kinect with thered markers. Com column refers to the command name which is either R (Right), L(Left), F (Forward), or S (Stop). T refers to total number**

Com	Microsoft Kinect for Windows				Red Marker System			
	T	Success	Failed	Success (%)	T	Success	Failed	Success (%)
R	300	280	20	93.3%	300	292	8	97.3%
L	400	280	120	70%	400	395	5	98.8%
F	400	240	160	60%	400	390	10	97.5%
S	200	200	0	100%	200	200	0	100%
T	1300	1000	300	76.9%	1300	1277	23	98.2%

Fig. 5 shows the centroid of the red markers compared with the centroid of the user’s hands using Mahalanobis distance for skin detection process [15]. The centroid extraction process using red markers is more robust without problems even though the user’s forearm is visible.

As described, many techniques are conducted to control robots using depth sensors, such as Microsoft Kinect depth sensor. We compared the accuracy of the robot’s responsive using red markers and Microsoft Kinect. For comparing between them in the proposed system, we used the same strategy of recognizing the user’s hand motions. In the red markers case, the control unit recognizes the red markers movements, however in the Microsoft Kinect case, the control unit recognizes the user’s hand movements using Kinect skeletal tracking [8] instead. The user held the red markers and performed a set of the predefined signs in the front of Microsoft Kinect instead of using web-cam.

The live-stream of the RGB camera in Microsoft Kinect was saved to be applied again to the proposed system in an off-line mode. For Microsoft Kinect, red markers have no usage, but they were used for evaluating the proposed system against Microsoft Kinect using the same input. Table 1 shows the result of the comparison using 1300 different users’ hand motions from the predefined signs, where our proposed system using Microsoft Kinect achieves 80% success responses from the robot, however the proposed system using red markers achieves 90% success responses from the robot, see Fig. 6.

### 3.2 Applications

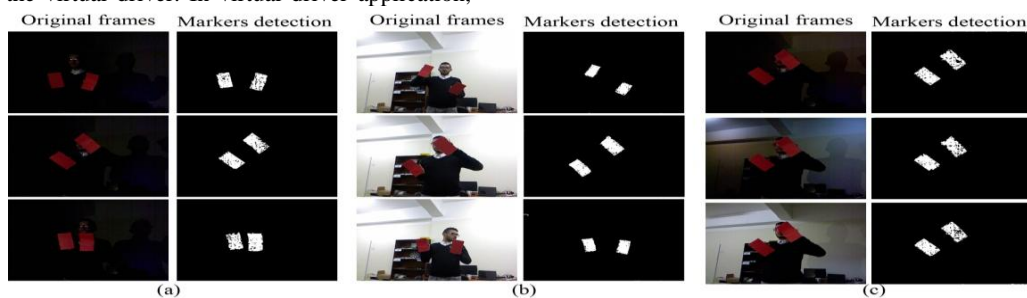
The proposed system has several applications, such as control robots in an easy and interactive way, manage a team of robots by controlling the master robot via user’s hand motions, and the virtual driver. In virtual driver application,

we can use the proposed system for visualizing car driving using the same hand movements of the driver for controlling a car. Beginner drivers can test their ability of controlling the car using the proposed system with some additions. We can use two photocells as car brakes and accelerator as shown in Fig. 7 that visualizes the application of the proposed system as virtual car driver. We can use the red marker strategy to ensure that the driver holds the steering in a sound manner (the two hands hold the steering), where the robot moves forward when the red markers that are attached to the hand’s proximal phalanges are visible according to the level of acceleration. The acceleration is determined based on the state of the visual accelerator.

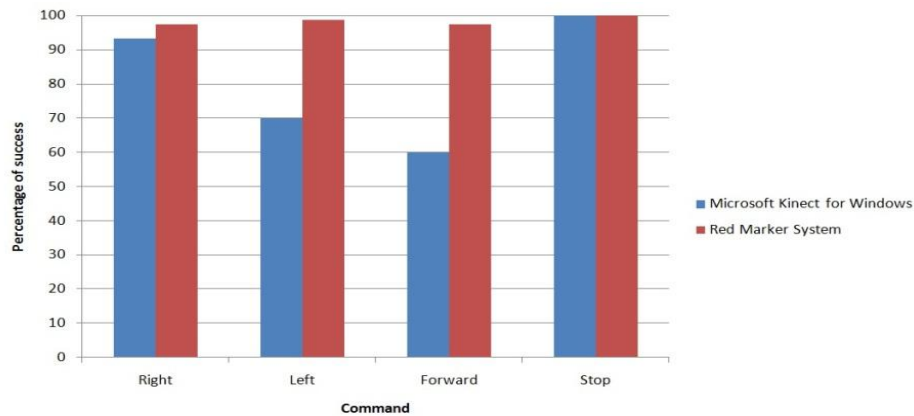
### 4. CONCLUSIONS

In this paper, a human-robot interaction system using human hand motions has been presented. The proposed system consists of two units (control unit and robot unit). In the control unit, the user’s hand motions are recognized for controlling a robot. The recognition process is performed using two red markers that are used for detecting the interesting area in each received frame from a web-cam in front of the user. After command recognition process, the required command is sent to the robot unit and the robot responds to the received commands.

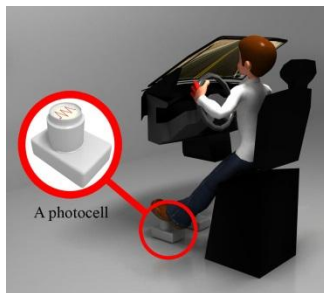
In addition, a web-cam streams what the robot sees to the user in the control unit. A virtual driver application has been presented as one of the proposed system’s applications. Where, the user can learn car driving in an interesting way. The proposed system achieved good results compared with controlling robots using Microsoft Kinect depth sensor using the same strategy of recognition of the user’s hand motions. So, the presented work introduces a low-cost human-robot interaction system.



**Fig 5: A comparison between the stability of the centroid using (a) red markers and (b) skin detection.**



**Fig 6: A comparison between the proposed system using Microsoft Kinect and thered markers.**



**Fig 7: A visualization of the virtual car driver application.**

## 5. REFERENCES

- [1] Spong M. W. and Fujita M. Control in robotics. The Impact of Control Technology: Overview, Success Stories, and Research Challenges (T. Samad and A. Annaswamy, eds.), IEEE Control Systems Society, 2011.
- [2] Monika Jain, AshwaniLohiyaAditi, Mohammad Fahad Khan, and AbhishekMaurya. Wireless gesture control robot: An analysis. Interna-tional Journal of Research in Computer and Communication Engineering, 1(10), 2012.
- [3] J Norberto Pires. Robot-by-voice: experiments on commanding an industrial robot using the human voice. Industrial Robot: An International Journal, 32(6):505–511, 2005.
- [4] J-B Gomez, AlexànderCeballos, FlavioPrieto, and TanneguyRedarce. Mouth gesture and voice command based robot command interface. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pages 333–338. IEEE, 2009.
- [5] Liu Zheng, Liu Yuliang, and Bing Zhigang. Vision-based human-computer interaction utilized in the robot control. In Intelligent Systems, 2009. GCIS'09. WRI Global Congress on, volume 2, pages 155–158. IEEE, 2009.
- [6] Min-Chi Kao and TS Li. Design and implementation of interaction system between humanoid robot and human hand gesture. In SICE Annual Conference 2010, Proceedings of, pages 1616–1621. IEEE, 2010.
- [7] PradeepShenoy, Kai J Miller, Beau Crawford, and Rajesh PN Rao. Online electromyographic control of a robotic prosthesis. Biomedical Engineering, IEEE Transactions on, 55(3):1128–1135, 2008.
- [8] Zhengyou Zhang. Microsoft kinect sensor and its effect. MultiMedia, IEEE, 19(2):4–10, 2012.
- [9] Michal Tölgyessy and Peter Hubinský. The kinect sensor in robotics education. In Proceedings of 2nd International Conference on Robotics in Education, pages 143–146, 2011.
- [10] Qingyu Li and Panlong Yang. Keep up with me: A gesture guided moving robot with microsoftkinect. In Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference on, pages 435–436. IEEE, 2013.
- [11] Guanglong Du, Ping Zhang, Jianhua Mai, and Zeling Li. Markerlesskinect-based hand tracking for robot teleoperation. Int J Adv Robotic Sy, 9(36), 2012.
- [12] Wei-chao Chen. Real-Time Palm Tracking and Hand Gesture Estimation Based on Fore-Arm Contour. PhD thesis, Master dissertation, Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, 2011.
- [13] Hui-Shyong Yeo, Byung-Gook Lee, and Hyotaek Lim. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. Multimedia Tools and Applications, pages 1–29, 2013.
- [14] Michal Choras'. Ear biometrics based on geometrical feature extraction. Electronic letters on computer vision and image analysis, 5(3):84–95, 2005.
- [15] Son Lam Phung, AbdesselamBouzerdoum, and Douglas Chai. A novel skin color model in ycbcr color space and its application to human face detection. In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages 1–289. IEEE, 2002.