Robotica (2015) volume 33, pp. 141–156. © Cambridge University Press 2014
doi:10.1017/S0263574714000083

Tracking and replication of hand movements by teleguided intelligent manipulator robot

A. T. Hussain*, S. Faiz Ahmed and D. Hazry

Autonomous & Machine Vision Research Cluster, Universiti Malaysia Perlis (UniMAP), Malaysia

(Accepted December 24, 2013. First published online: February 11, 2014)

SUMMARY

In this paper, a new method is presented that allows an intelligent manipulator robotic system to track a human hand from far distance in 3D space and estimate its orientation and position in real time with the goal of ultimately using the algorithm with a robotic spherical wrist system. In this proposed algorithm, several image processing and morphology techniques are used in conjunction with various mathematical formulas to calculate the hand position and orientation. The proposed technique was tested on Remote teleguided virtual Robotic system. Experimental results show that proposed method is a robust technique in terms of the required processing time of estimation of orientation and position of hand.

KEYWORDS: Intelligent manipulator; 3D Hand tracking; Human computer interaction; Morphology.

1. Introduction

Vision-based tracking of hand motions is currently a very active area of research e.g. Pavlovic,[1] Maung,[2] Manresa,[3] Elmezain.[4] However, the problem of correctly identifying a hand and completely simulating its movement is far from being solved. A number of previously proposed algorithms extract features from the hand and classify it by using previously obtained information. Examples of these techniques include the use of Neural Networks by Maung,[2] Genetic Algorithms by Cui,[5] Hidden Markov Models by Binh,[6] Lee[7] and others Starner,[8] Kjeldsen,[9] Zhao,[10] Yoon,[11] Lockton.[12] Although some of these techniques have reported certain levels of success but these solutions suffer from the restrictions of preset gestures and positions, which are fed into the system beforehand. Furthermore, several computer solutions that are already available for tracking human hands require invasive add-ons, such as gloves or targets by Bowden.[13] This paper describes a method to identify the human hand in real time and analyze the resultant information to obtain the hand's current position and orientation without above mentioned restrictions. Although plain background is used in this research work, the results obtained showed high consistency with hands of different sizes and colors and the simulated movement and rotation matched that of the subject's hand.

After obtaining and matching the hand's orientation and position in 3D space, the results are fed into a simulated robotic manipulator and the position and orientation of its end-effector accurately matched the user's hand. Furthermore, additional information from the hand, such as the opening or closing of the end-effector gripper, is obtained and directly sent to the simulated manipulator robot, which emulated the action.

2. Literature Review of Hand Gesture Recognition

A large amount of research work has been performed on hand gesture recognition. Some of the most recent achievements include the work by Moon-Jin Jeon Bien,[14] who constructed a system with a reliable and robust hand gesture recognition algorithm that used an efficient control-based fuzzified decision making technique but faced difficulties when using a multivariate fuzzy decision tree. George Caridakis's[15] recognition scheme for hand gestures utilizes self-organization feature

* Corresponding author. E-mail: asthussain@yahoo.com

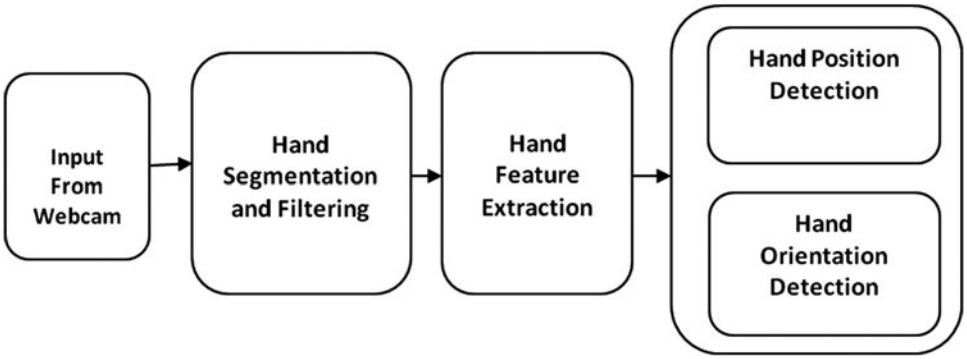

Fig. 1. Block diagram of proposed algorithm.

maps to model information extracted through image processing. This scheme consisted of a lengthy process that built two models for each gesture category, along with the appropriate distance metrics, using self-organizing Markov maps. In 2010, Heung Il Suk *et al.*[16] built a model that recognized hand gestures in a continuous video stream using a dynamic Bayesian network, or DBN, model, in real time. However, the implementation was complicated because it required extensive analysis of hand shapes. Tin Hninn Hninn Maung[2] developed a robust recognition system for Myanmar Alphabet Languages gestures in 2009. Although users do not need to wear gloves and a uniform background or special hardware is not required, this tracking method was limited to 2D. Earlier works have several more limitations and are generally found to exhibit as lower processing speed.

2.1. Proposed algorithm for 3D hand gustier recognition & estimation

For the recognition of 3D human hand gustier, proposed algorithm follows the block diagram shown in Fig. 1. The raw image data from webcam first goes through the hand segmentation and filtering processes and then hand features are extracted by using morphology approach and finally hand position and its orientation are determined. The detail explanations about these processes are described in following section.

3. Hand segmentation

Before determining the position and orientation of the hand from webcam image, we need to segment and identify the human hand within the image.

The process of hand segmentation is presented in the flow chart shown in Fig. 2. The input video which is ran at 30 frames per seconds at a resolution of 640×480 pixels, first converted from the RGB spectrum into a grayscale image. This is accomplished by simply averaging the Red, Green and Blue intensity values in each of the pixels, which results in a matrix with the number of rows and columns matching the number of pixels in the snapshot.

The Sobel edge detection method is used as a filter to process the grayscale image. After the image passes through the filter, its gradient magnitude, G, (shown in Eq. 3) is computed as the square root of the sum of the square of each vertical filter, which is shown in Eq. (2), and the horizontal filter, which is shown in Eq. (1):

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \text{image}, \tag{1}$$

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \text{image}, \tag{2}$$

$$G = \sqrt{\left(G_x^2 + G_y^2\right)}, \tag{3}$$

where the symbol '*' denotes the convolution operator.

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

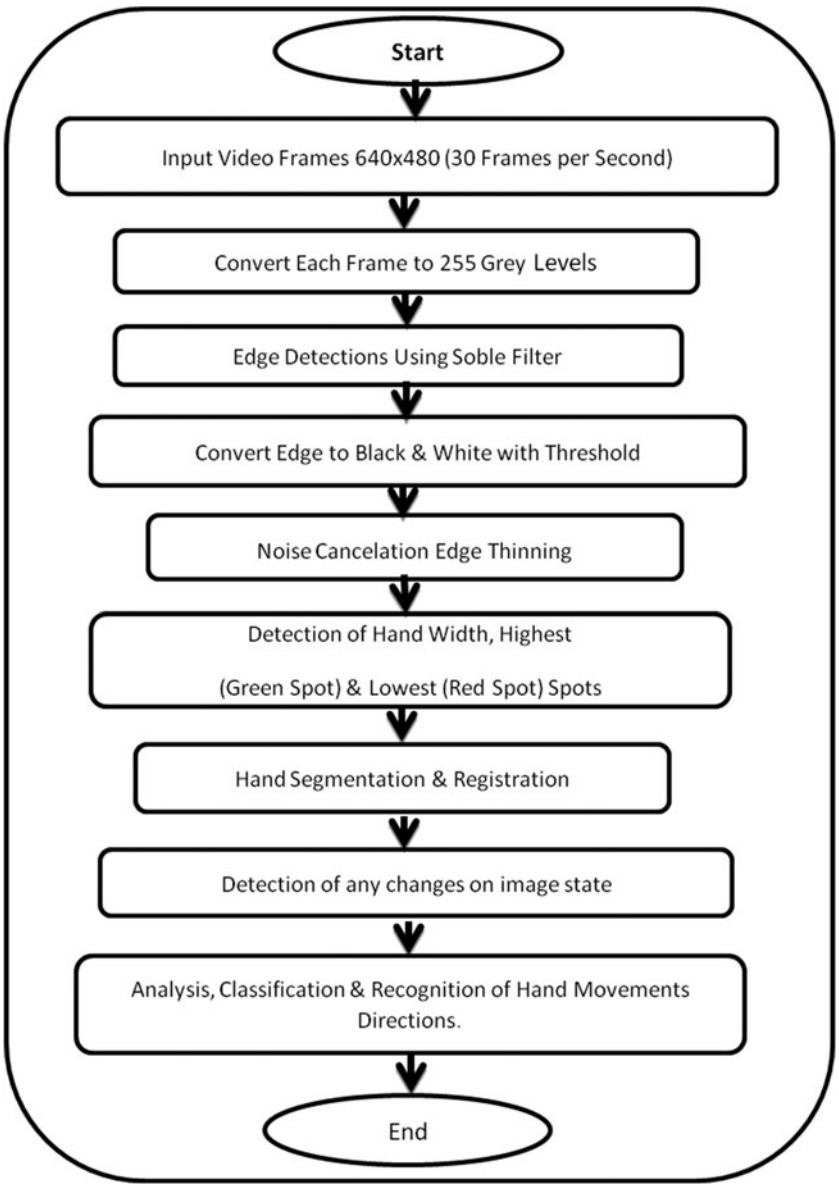

Fig. 2. Flow chart of hand Segmentation process.

The result is a grayscale image with high intensity at the border of the hand and fingers and at any edges or marks on the palm or the outside face of the hand. A simple thresholding operation is performed to transform this image into a black and white Boolean image, in which each pixel was classified as a white-intense-object or a black-not intense-background.

The flexibility of changing the threshold provides better adaptability to different lighting conditions, different skin tones or other similar differences. Because the background is white, thresholding at a lower intensity value can give good results that allow the algorithm to continue to detect further hand features. The end result is a binary image matrix that identifies the hand border edges as well as some edges inside the hand. Because the edges inside the hand border are inconsequential, there is no need to sacrifice any additional processing in their analysis. However, some noise may be introduced in areas outside the hand border. This noise can be eliminated by removing the small number of pixels that represent these objects in the image using a morphological opening operation.

The morphological opening is accomplished by first scanning through all the pixels in the image and recording those that are non-zero. This is followed by the use of a union-find connected components algorithm[14] to identify those pixels that belong to each of the objects in the image. After all the

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

objects in an image have been identified, their areas are calculated by simply counting the number of pixels in each. Those objects with a pixel area lower than the threshold are then identified and the corresponding pixel values are inverted, which, in essence, means that the objects are removed.

The final image processing step is the morphological thinning operation of the hand object in the image. The thinning algorithm is as follows:[26,27]

- Divide the image into two distinct subfields in a checkerboard pattern.
- In the first subiteration, delete pixel p from the first subfield if and only if conditions G_1, G_2, and G_3 are all satisfied.
- In the second subiteration, delete pixel p from the second subfield if and only if conditions G_1, G_2, and G_4 are all satisfied.

Condition G_1:

$$X_H(p) = 1, \tag{4}$$

where

$$X_H(p) = \sum_{i=1}^{4} b_i \text{ with } b_i = \begin{cases} 1, & \text{if} \quad x_{2i-1} = 0 \quad \text{and} \quad (x_{2i} = 1 \; or \; x_{2i-1} = 1) \\ 0, & \text{Otherwise} \end{cases}$$

where $x_1, x_2, \ldots, x_8$ are the values of the eight neighbors of p, starting with the east neighbor and numbered in a counter-clockwise order.

Condition G_2:

$$2 \leq \min\{n_1(p), n_2(p)\} \leq 3, \tag{5}$$

$$\text{where} \begin{matrix} n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k} \\ n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1} \end{matrix}$$

Condition G_3:

$$(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0. \tag{6}$$

Condition G_4:

$$(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0. \tag{7}$$

The two sub iterations combined form a single iteration of the thinning algorithm. The iterations are repeated until the image stops changing. The results from these hand segmentation steps are shown in Fig. 3.

4. Extraction of hand features

After the hand segmentation, the boundary pixels of the hand and some of the interior pixels have been determined. To identify the hand position in a 3D space, it is necessary to pinpoint a reference point in the hand. Therefore, the hand reference point chosen was the middle of the wrist joint, as shown in Fig. 4.

The reference point can determine by scanning horizontal width of the hand for each point in the vertical axis. The horizontal line that passes through the reference point is significantly shorter in width than the nearby horizontal lines. The algorithm therefore determines the location where a jump in the horizontal width of the segmented hand occurs; this provides a cue for the identification of the wrist area.

Proposed algorithm separates the vertical height of the image into designated horizontal lines and assumes that the reference point lies on one of these lines. This causes the formation of some sort of dead-bands, where changes in the vertical position of the reference point are only taken into account

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

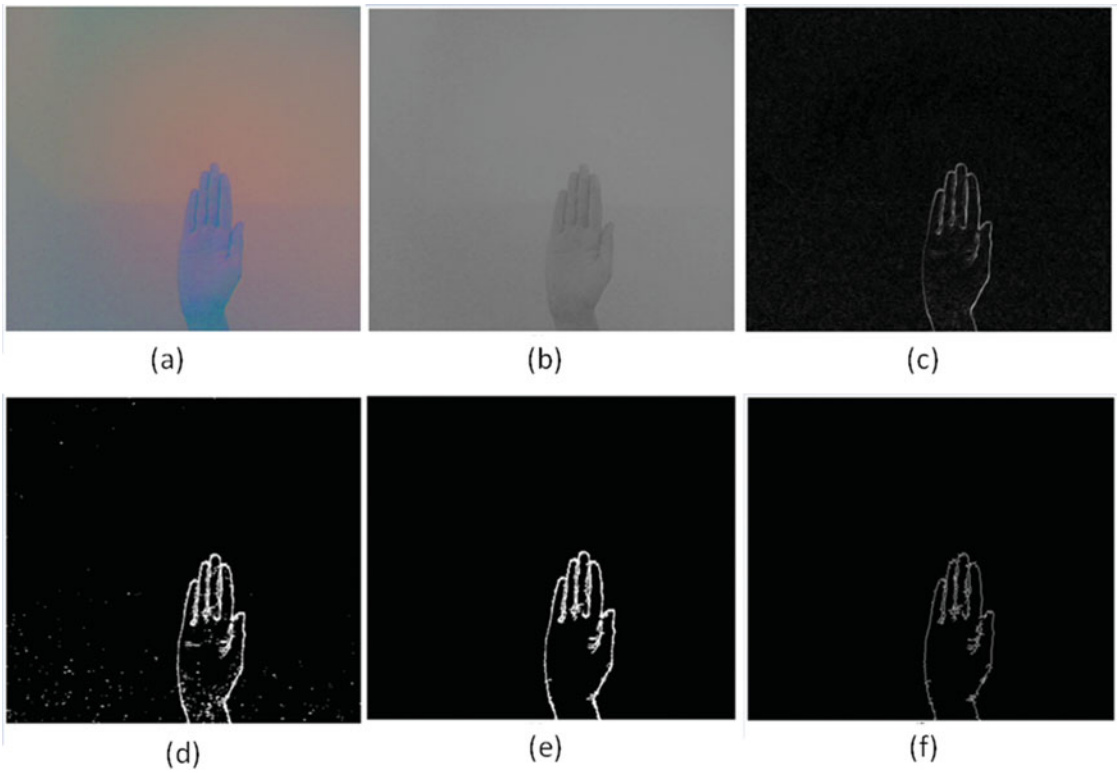

Fig. 3. (Colour online) Image progression of hand segmentation : (a) original snapshot, (b) after conversion to grayscale, (c) filtered image, (d) after conversion to black and white, (e) after noise removal and (f) after thinning.

Fig. 4. (Colour online) Proposed reference point for hand tracking.

if they are deemed large enough and small changes do not contribute to the location of the vertical position, as shown in Fig. 5.

The problem lies in the identification of the vertical position of the reference point because, when the hand is rotated, the wrist joint ceases to be horizontal. Consequently, the scanning procedure should not be performed on horizontal lines perpendicular to the vertical scanning operation, but at a slight angle. Therefore, once the rotation about the *y*-axis has been found using a preliminary estimation of the reference point, an image rotation algorithm rotates the hand to the original rotation about the *y*-axis, thereby cancelling the effects of the rotation. Consequently, the image is maintained at the upward perpendicular position and the algorithm calculates the biggest width difference to determine the value of the hand's vertical position, as shown in Fig. 6.

After obtaining the vertical position, the horizontal position of the reference point is calculated as the mid-point between the two ends of the wrist joint. This reference point is then used to accurately

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

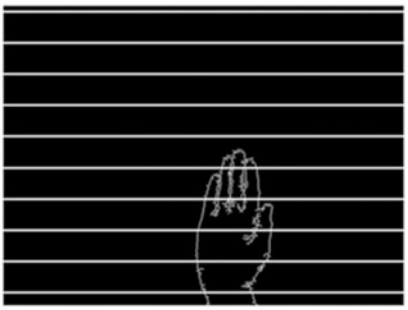

Fig. 5. The image is separated into zones and the vertical position of the reference point lies on only one of these zones.

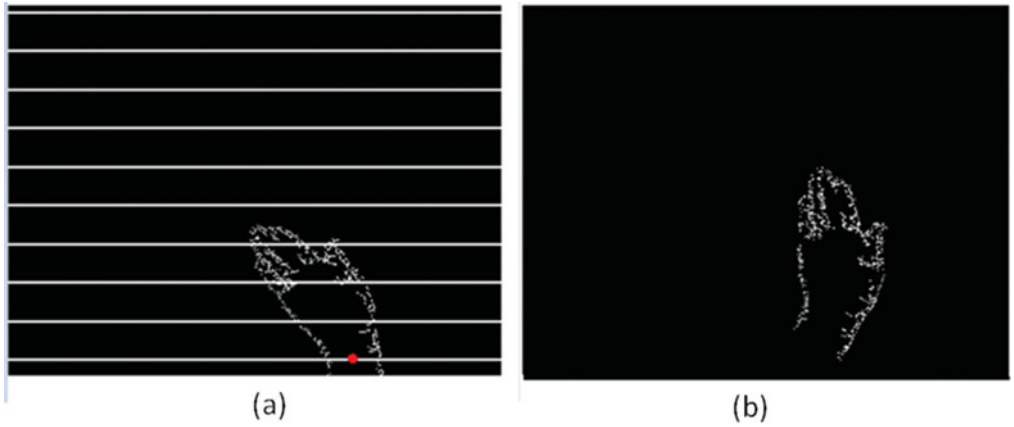

Fig. 6. (Colour online) (a) Normal view of the hand (red spot indicates the reference point), (b) Because the rotated hand gives a false premise for the reference point, its rotation is reversed to obtain a more accurate view of the width of the hand.

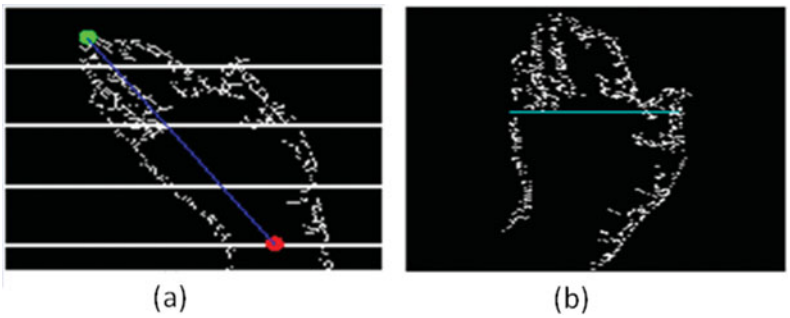

Fig. 7. (Colour online) (a) Many features were extracted from the segmented human hand image, including height, length and position, and width. (b) Example of width measure.

obtain the horizontal and vertical position of the hand in 3D space. The next feature determined by the algorithm is the length of the hand. The farthest point on the hand from the reference point is therefore identified. This calculation is performed by obtaining the distance in both x and y between each of the pixels in the hand and the reference point. The pixel that has the largest distance magnitude will be declared the farthest point and thus the tip of the hand. Consequently, the algorithm has identified the most important features of the human hand, most notably its length, height, width and vertical and horizontal positions. These features, which are noted in Fig. 7, will then be used to obtain the position and orientation of the human hand in 3D space.

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

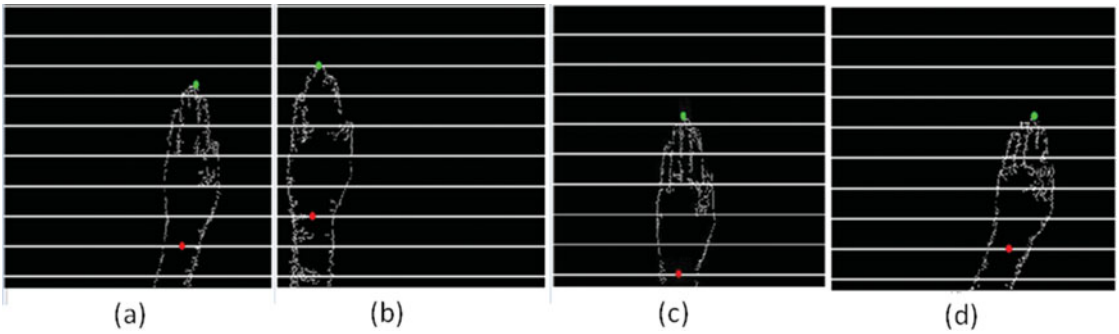

Fig. 8. (Colour online) The different x and y coordinates of the hand can be easily tracked. Examples are shown of the detection of (a) left hand side movement, (b) right hand up and side movements, (c) left hand backward movement and (d) left hand side and backward movements.

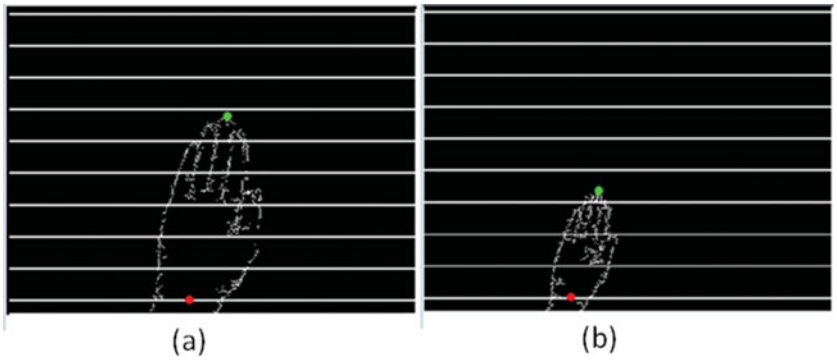

Fig. 9. (Colour online) Comparison of identification features after two successful hand registrations.

5. Estimation of Hand Position

After the x and y coordinates of the reference point are identified, the position of the hand in these two directions can be obtained, as shown in Fig. 8.

A number of methods that share the same general structure are proposed for tracking the z coordinate of the hand position in 3D space (indicated by moving nearer to or farther from the camera). Often, at the beginning of the recognition process, a number of values are obtained and saved for future reference in a process called registration. These values, which are extracted in real-time and stored in a memory location, consist of important measurements, such as the hand length, hand width, and the number of pixels (indicating area), and are termed identification features.

The correct method to initialize this registration process lies in detecting the moment the hand is idle for a certain amount of time. This is performed by determining the change in the position and orientation of the hand after each scan. When the change in these values is very low for a consecutive number of scans or frameshots, the hand is deemed idle and its identification features are calculated. This registration process is performed every time the hand is idle in front of the camera. The identification features registered after each registration are compared to the values obtained from previous registration occurrences. The comparison of the key features, such as new hand width and length with the previous values, allows the identification of the position of the hand in the z direction (near or far from the camera) through the use of simple relativity mathematics.

An example of this procedure is shown in Fig. 9. Notice that the hand lies idle in the initial run to register the initial hand features, such as hand width and length. After the hand is moved away from the camera (in the negative z direction), it remains idle for another period of time, at which point the new length and width of the hand are calculated. The comparison of these new properties with the old values obtains a measure of the z coordinate of the hand position in 3D space.

Because the hand in Fig. 9(b) is half as long the hand in Fig. 9(a), the hand in Fig. 9(b) is twice as far away from the camera than in Fig. 9(a). Although the wait period involved in using this method means that its execution is not in real-time, it is worth noting that the position in this third dimension

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

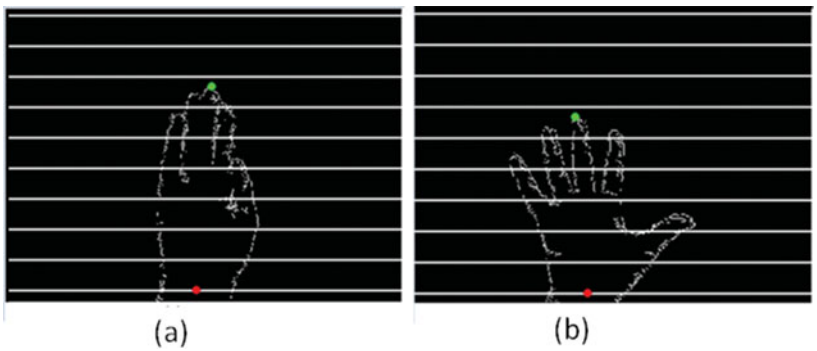

Fig. 10. (Colour online) (a) Normal hand width. (b) When the maximum hand width increases significantly, the gripper hand status is changed from closed to open.

is of no real importance and its value can be ignored. The wait and compare method can be replaced by a real-time method of obtaining the width or length of the hand at each scan and comparing it to the one before. Although this proposed real-time method produces accurate results of the hand position in all three planes, it experiences problems when obtaining the value for one of the orientation vectors of the hand. Thus, a choice has to be made in the real time computation of either the z position or one of the orientation vectors, which explains the partial real-time nature of the z coordinate calculations.

During the computation of the hand position, another feature can be obtained, namely the open or closed state of the gripper. This feature can be easily triggered by a cue from the user's hand that would act as a signal in the algorithm. A practical cue, which was used in this algorithm, was a rapid change in the hand width. The algorithm was set to automatically detect any change in the width larger than 150% of the normal hand widths in the same z coordinate. If this sudden increase is detected, the robot manipulator's gripper will switch from closed to open. The same is true for the opposite change (from open to closed). An example of this procedure is shown in Fig. 10.

6. Deduction of 3D Hand Orientation

After the hand position has been estimated, the same set of features can be used to deduce the orientation of the hand in 3D space. The three quantities that would need to be determined are the angles that the hand has rotated with respect to the three principal axes, namely, the x, y and z-axes.

The quantity that can be most simplyobtained is the rotation about the z-axis (depth), which occurs on the plane parallel to the screen. This angle is measured by using two previously obtained features: the reference point (located in the middle of the wrist) and the tip of the hand (farthest point on the hand from the reference point). After obtaining the coordinates for these points on the plane parallel to the screen (the xy plane), a line can be drawn between them. The angle that this line makes with the horizontal x-axis, which can be computed through the use of the atan2 function, is the angle the hand is rotated about the z-axis.

The atan2 function is similar to the arctan function except that it correctly identifies the quadrant of the rotated angle, which means that it is possible for the human hand to rotate more than 90°. In practice, the rotation of a hand lies in the range of $0^\circ - 180^\circ$. The atan2 function is defined in Eq. (8):

$$\text{atan2}(b,a) = \begin{cases} \arctan\left(\frac{b}{a}\right), & a > 0 \\ \pi + \arctan\left(\frac{b}{a}\right), & b \geq 0, a < 0 \\ -\pi + \arctan\left(\frac{b}{a}\right), & b < 0, a < 0 \\ \frac{\pi}{2}, & b > 0, a = 0 \\ \frac{-\pi}{2}, & b < 0, a = 0 \\ \text{Undefined}, & b = 0, a = 0 \end{cases} \tag{8}$$

where the value of b is equal to the difference between the x coordinate of the reference point and the x coordinate of the human hand tip. Similarly, the value of a is equal to the difference between the y

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

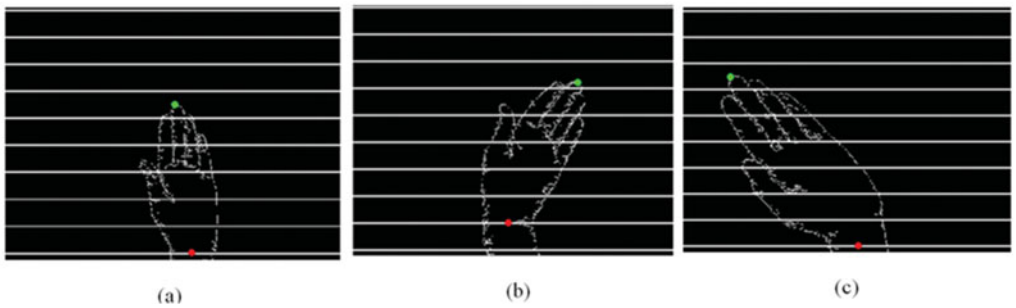

Fig. 11. (Colour online) Rotation of the human hand about the z-axis: (a) angle of rotation approximately 90°, (b) angle of rotation less that 90° and (c) angle of rotation greater than 90°.

coordinate of the reference point and the y coordinate of the hand tip. Using these calculations, the first angle of the orientation of the human hand is obtained, as shown in Fig. 11.

The calculation of the next two angles required to complete the detection of the orientation of the hand is more difficult. The first of these, the angle about the y-axis (the vertical axis) utilizes the hand width property, which was previously obtained. As noted in the previous sections, the hand width is recorded at the beginning of the algorithm and any subsequent change in this property is noted and compared to the original registered width. The method of accurately measuring the hand width at any given moment as it is rotated about the various axes has already been presented. A decrease in the hand width in successive scans (frame shots) indicates rotation about the vertical y-axis.

By assuming that the original registration of the human hand was calculated with a 0° rotation about the y-axis, then the width at this initialization stage is the maximum value. Therefore, each decrease in the hand width reflects a relative rotation about the y-axis. If it is assumed that a full rotation of 90° results in a hand width of 1/2 the maximum hand width, a relationship between the angle and the width can be formed, as shown in Eq. (9):

$$\theta_y = \cos^{-1}\left(2\left(\frac{w}{w_o} - \frac{1}{2}\right)\right), \tag{9}$$

where w is the instantaneous hand width and wo is the original registration of the human hand width. To minimize the effect of trembling or a sudden change in angle, it is suggested that these angle values be rounded to the closest 5°. Another approach that could be used considers that the original registered human hand width was calculated at an angle other than 0°. In this case, the hand's rotation about the y-axis can be classified in two directions; clockwise and counter-clockwise. In this case, an increase in the hand means a rotation in one direction, whereas a decrease corresponds to a rotation in the opposite direction.

We should note that a fully rotated hand about the y-axis (at 90° rotation) can have a high percentage of width variation in relation to the original full width. This variation, which differs between humans and even between different hands (right or left), can be taken into account by changing the equation that relates the width to the rotation about the y-axis. However, as with most of the other variables presented earlier, the different hands used in our experiments do not exhibit a significant difference in the parameter values that are used in Eq. (9). This is especially true if the angle values are rounded to the nearest 5°. Figure 12 shows an example that demonstrates the algorithm used to determine the rotation about the y-axis.

In addition, we need to obtain the rotation angle of the human hand about the x-axis. The approach used to obtain this value is very similar to that used to obtain the rotation about the y-axis; the only difference is the use of the hand length parameter instead of the width. The comparison of the length obtained from the initial registration with the length after each time interval can provide an indication of the human hand's rotation about the x-axis.

The initial registration position (from which the original length is calculated) determines which set of angles correspond to the human hand's orientation in future scans. For example, if the original length is a certain value, then a larger length corresponds to a decrease in the rotation angle about the x-axis (clockwise), whereas a shorter hand length reflects an increase in the angle. Alternatively,

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

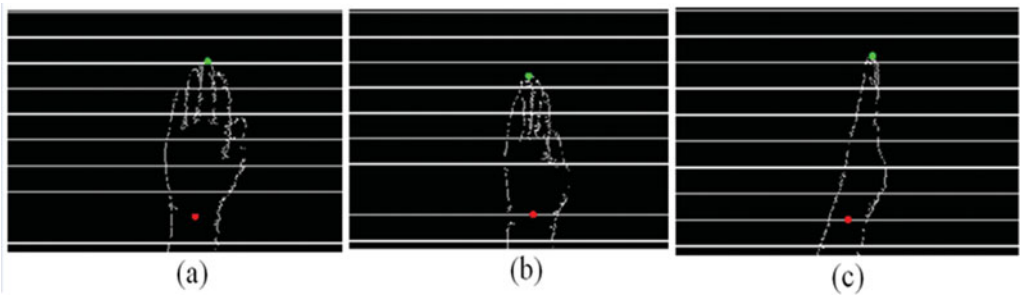

Fig. 12. (Colour online) Differing hand widths correspond to rotations about the y-axis: (a) no rotation (0°), (b) partial rotation and (c) full rotation (90°).

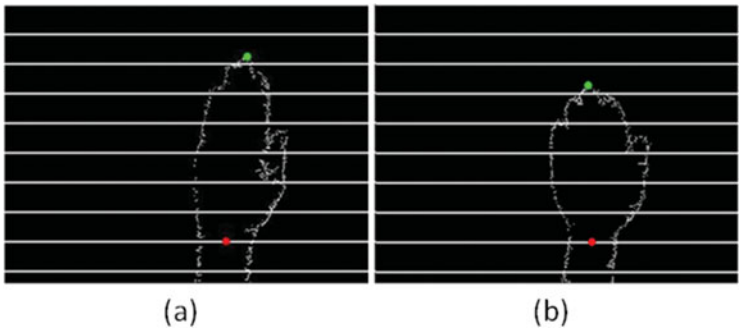

Fig. 13. (Colour online) Rotation about the x-axis. (a) Original reference image. (b) A reduction in the human hand length corresponds to an increase in the rotation angle.

if the length is considered to be maximal in the initial registration, this length cannot increase and thus the angle can only decrease. This gives a range of rotation of approximately 90°, which is a realistic range for the possible rotation of the human hand about the x-axis. An example of this part of the orientation detection algorithm is shown in Fig. 13. Additionally, Fig. 14 shows the effect ofchanging the algorithm's edge detection intensity threshold, whereas Fig. 15 shows an assortment of hand configurations and the corresponding identification by the algorithm of the position and rotation about the three axes.

Figure 15(a) is the original registration, the hand width and length are determined and recorded. Figure 15(b) shows an increase in the y position, a decrease in the x position and a constant hand length and width (after image rotation) means that there is no rotation about the x and y-axes and an increase in the rotation about the z-axis. Figure 15(c) shows an increase in the x position, a decrease in the y position, and a decrease in the hand width corresponds to no rotation about the x-axisand increased rotation about the z and y-axes.

7. Robotic Wrist Manipulation

The accurate hand position and orientation in 3D space are fed into the robot manipulator to control the position and orientation of the robotic end-effector such that it matches the hand movements. The robotic manipulator used in this study is a modified SCARA robot. The proposed design is a SCARA manipulator with a spherical wrist as its end-effector, which gives it complete freedom of movement in 3D space with regards to position and orientation.

Each of the links of the robotic manipulator is assigned to a coordinate frame. The Denavit and Hartenberg (DH) convention proposes a matrix method of systematically assigning coordinate systems to each link of an articulated chain. The axis of revolute joint i is aligned with axis $zi - 1$. The $xi - 1$ axis is directed along the normal of axes $zi - 1$ to zi for intersecting axes parallel to $zi - 1 \times zi$. The link and joint parameters may be summarized as follows:

- Link length a_i is the offset distance between axes z_{i-1} and z_i along the x_i-axis;
- Link twist α_i is the angle from the z_{i-1}-axis to the z_i-axis about the x_i-axis;

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

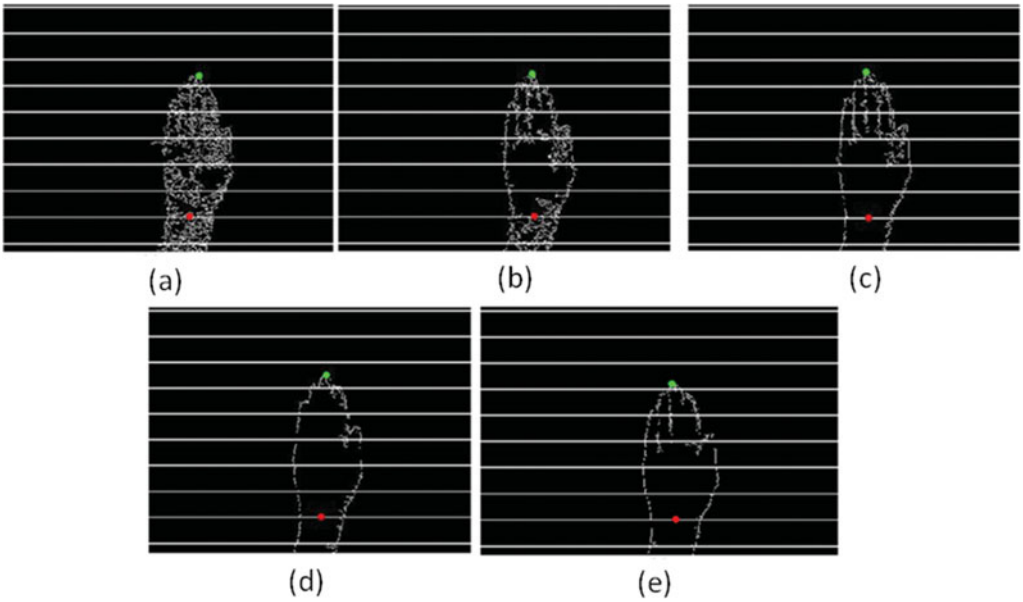

Fig. 14. (Colour online) Effect of a Changein the intensity thresholdto: (a) 7.5, (b) 10, (c) 12.5 (default), (d) 15 and (e) 17.

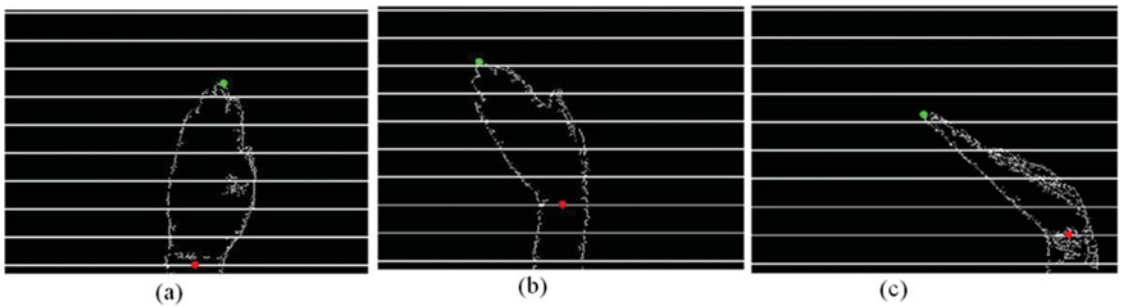

Fig. 15. (Colour online) Different human hand configurations.

- Link offset d_i is the distance from the origin of frame $i-1$ to the x_i-axis along the z_{i-1}-axis;
- Joint angle θ_i is the angle between the x_{i-1} and x_i-axes about the z_{i-1}-axis.

These parameters and conventions are applied with our robotic manipulator. The resultant DH parameters are shown in Table I.

Using these parameters, the inverse kinematics model describing each link angle and link length with respect to the desired position and orientation of the manipulator wrist can be calculated using Eqs. (10) and (11):

$$d_3 = L_1 - z_{\text{des}}, \tag{10}$$

$$\theta_2 = \pm \cos^{-1}\left(\frac{p_x^2 + p_y^2 - L_3^2 - L_2^2}{2L_2L_3}\right), \tag{11}$$

where z_{des} is the desired orientation and position of the robot arm, which should be identical to the human hand position and orientation.

After obtaining θ_2, the system for $\sin\theta_1$ and $\cos\theta_1$ shown in Eq. (12) can be solved:

$$\begin{bmatrix} -L_3\sin\theta_2 & L_2 + L_3\cos\theta_2 \\ L_2 + L_3\cos\theta_2 & L_3\sin\theta_2 \end{bmatrix} \begin{bmatrix} \sin\theta_1 \\ \cos\theta_1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}, \tag{12}$$

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

Table I. DH parameters for the proposed simulated robotic manipulator links.

Link (i)	α_{i-1}	a_{i-1}	d_i	θ_i	q_i(Variable)
1	0°	0	L_1	θ_1	θ
2	0°	L_2	0	θ_2	θ
3	180°	L_3	d_3	0	d
4	−90°	0	0	θ_4	θ
5	90°	0	0	θ_5	θ
6	−90°	0	0	θ_6	θ

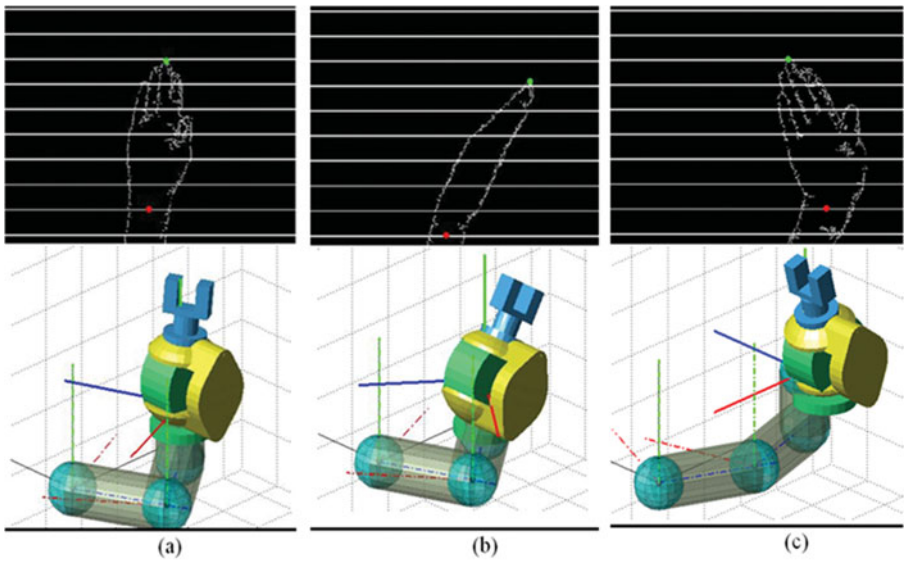

Fig. 16. (Colour online) Simulation of the configuration of a human hand with a robotic manipulator: (a) original registered configuration, (b) rotation about two angles and (c) rotation and translation.

where $\theta_1 = \text{atan2}\,(\sin\theta_1, \cos\theta_1)$. The desired orientation (rotation) matrix uses the notation shown in Eq. (13):

$$R_{\text{des}} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{13}$$

The corresponding rotation matrices for the three rotation angles can then be multiplied, as shown in Eq. (14), to obtain R_{des}.

$$R_{des} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix}. \tag{14}$$

The orientation matrix can then be used to obtain the desired rotation angles, as shown in Eq. (15):

$$\theta_5 = \pm\text{atan2}\left(\sqrt{r_{21}^2 + r_{22}^2}, r_{23}\right), \quad \theta_4 = \pm\text{atan2}\left(\frac{r_{33}}{\sin\theta_5}, \frac{-r_{13}}{\sin\theta_5}\right), \quad \theta_6 = \pm\text{atan2}\left(\frac{-r_{22}}{\sin\theta_5}, \frac{r_{21}}{\sin\theta_5}\right). \tag{15}$$

Consequently, for any desired orientation and position of the robotic manipulator, we can obtain the link angles and lengths required. The matching of the state of the human hand in 3D space to a real robot can be easily accomplished through simple mathematical calculations. An example of this mapping is shown in Fig. 16.

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

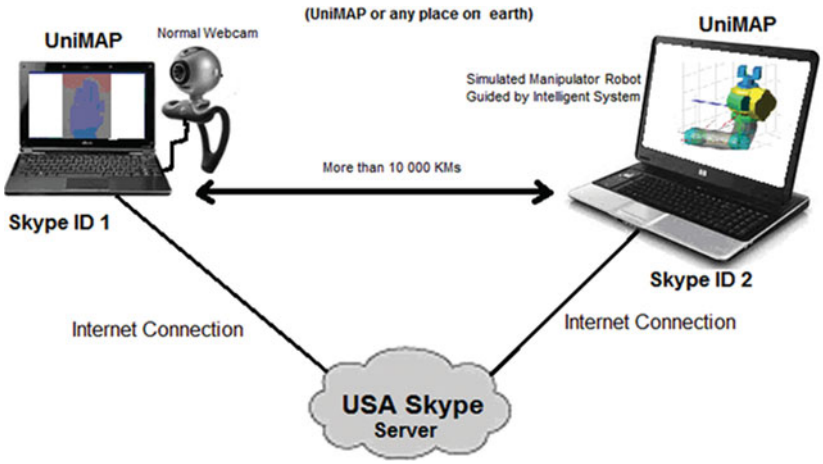

Fig. 17. (Colour online) Schematic diagram of the experimental setup of the teleguided manipulator robotic system over long distances.

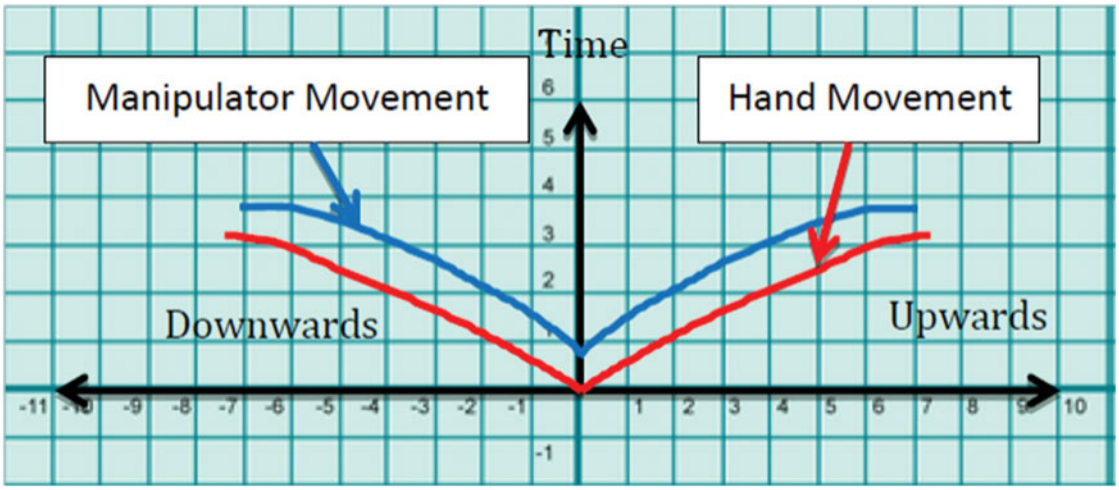

Fig. 18a. (Colour online) Response time of hand movement and robot manipulator in Left and Right movement (Response time is 0.88 s).

8. Application

This morphology intelligent system has been implemented and tested to teleguide a manipulator robot using an internet connection. We chose to use a Skype-based webcam connection, with a webcam at one end and, at the other end, a computer system with Skype, the morphology program and a simulated robot that replicates the hand movements recorded through the webcam. The computer processor was a Core-2-Duo at 1.8 GHz with 4 GB of RAM and the internet connection speed was 8 MBs-1. A total of two experiments were conducted with this configuration.

8.1. Experiment 1: Remotely teleguided intelligent manipulator system

We logged in from UniMAP, Malaysia using 2 Skype account IDs connected through a server located in the USA to ensure that the teleoperation occurs over a long global distance. The response inevitably depends on the average speed of the Internet connection experienced by both computers. Figure 17 shows a schematic of the experimental setup. Figures 18(a) and 18(b) shows the response of hand movement and robot manipulator movement. Several hand movements were executed at one end and the results obtained at the other end matched those hand movements with minor errors.

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

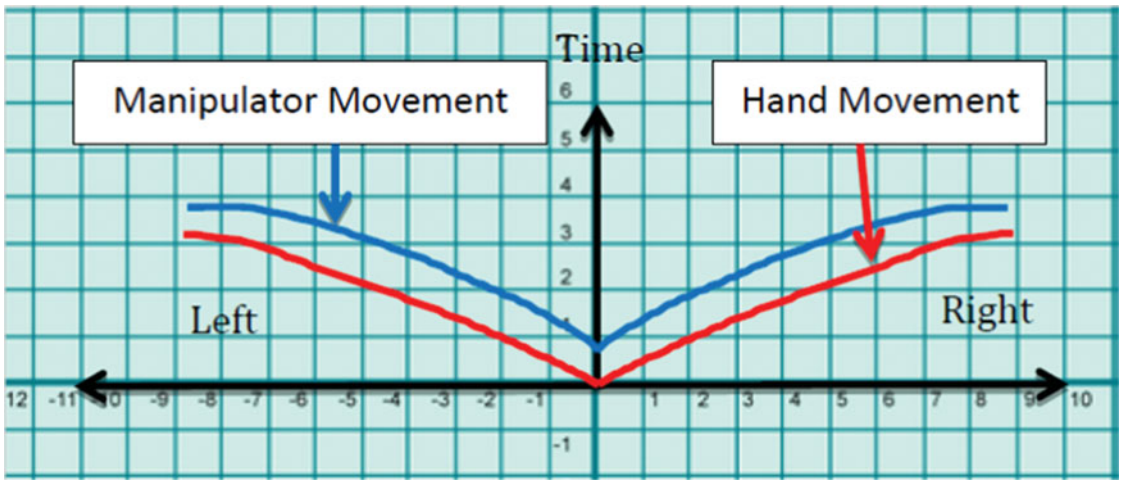

Fig. 18b. (Colour online) Response time of hand movement and robot manipulator in Up and Down movement (Response time is 0.9 s).

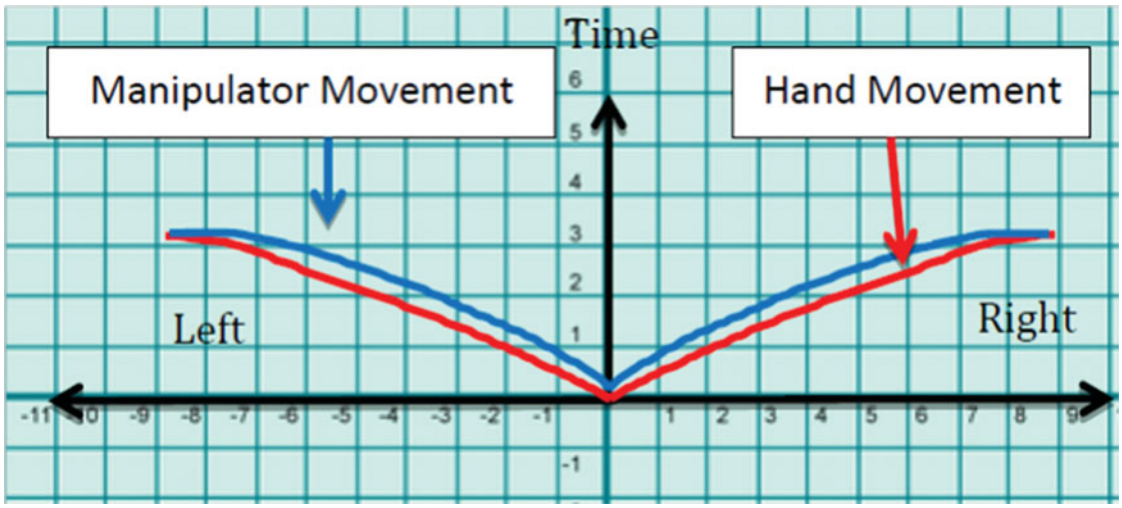

Fig. 19a. (Colour online) Response time of hand movement and robot manipulator in Left and Right movement (Response time is less then 0.2 s).

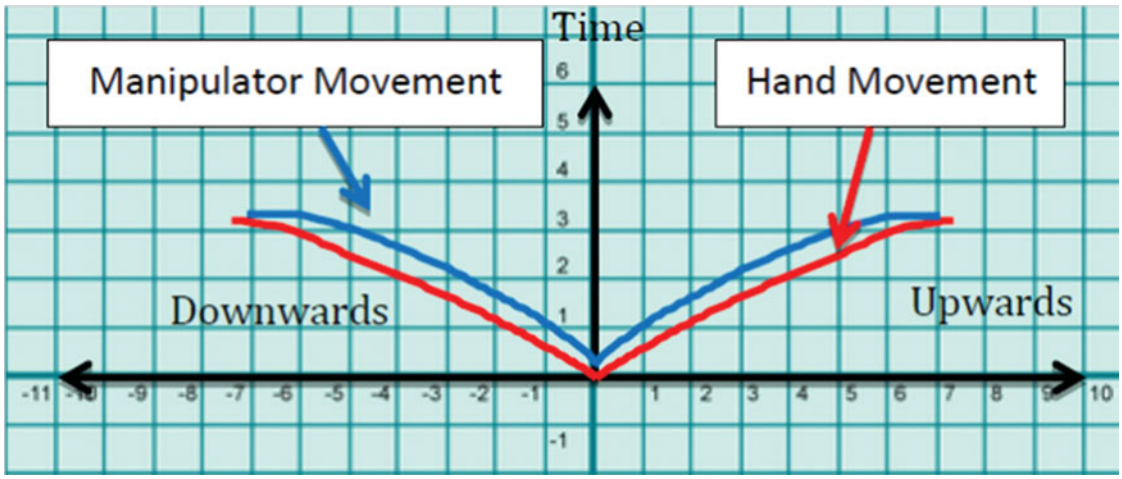

Fig. 19b. (Colour online) Response time of hand movement and robot manipulator in Up and Down movement (Response time is 0.3 s).

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

8.2. Experiment 2: Locally teleguided intelligent manipulator system

When the experiment was carried out within the same computer system, the accuracy of the results was similar but the reaction time was faster than in the previous experiment. Figures 19(a) and 19(b) shows response times for this experiment. The robotic manipulator moved in almost real time because the hand movement recording, the analysis of the position and orientation and the robotic manipulation occurred on the same machine.

9. Comparison of Proposed Algorithm with other work:

The advantages of proposed algorithm are comparison with other techniques as shown in table II.

Table II. Comparison between Proposed technique and other related work for hand gesture recognition and its implementation.

Proposed technique	Other techniques (Hybrid Fd's & bpnn)
1. Easy to perform analysis and measurements for the used objects.	1. Difficulties to perform analysis and measurements for the used objects.
2. It can be designed simply using available extracted information.	2. It can be designed using extracted information with many trials to confirm.
3. Always leads to easy implementation, using available hardware (no need for highly sophisticated equipment).	3. Always leads to not easy implementation, using available hardware (some time needs for highly sophisticated equipment).
4. Results are mostly in real time.	4. Results are mostly not in real time, due to the delay of used equipment.
5. Low cost effective to achieve system application and implementation.	5. Always a high cost needed to achieve system application and implementation.
6. It's flexible for further development.	6. It needs technology and techniquesfor further development.
7. It is applicable with any shape (close, open or including holes)	7. It is sensitive open holes inside the shape
8. It has a high perforce of recognizing unknown different objects on the same category or different categories	8. It has low performance of recognizing unknown different objects on the same category, useful for recognizing a group of objects on the same category i.e. recognizing categories

10. Conclusion

In this paper, an algorithm has been proposed that correctly and accurately extracts the 3D position and orientation of a human hand. Then the results were fed to a simulated robotic manipulator directly or over internet working sites (long range distance of teleoperation). In all our experiments, the manipulator was successfully teleoperated. The proposed algorithm initially executes edge detection, extracts features, such as hand width and length, and used these to perform mathematical operations to identify the position and rotation of the hand. The developed algorithm was observed to produce instantaneous responses, although this depends on the internet connection speed and the computer specifications. When we test our system over the internet the robot manipulator took 0.8 to 0.9 second response time, but when we test same system on the single computer then robot manipulator quickly respond and it takes only 0.2 to 0.3 seconds. Experimental results proved that proposed algorithm is fast and robust due to its simple mathematical operations.

We conclude this paper with a summary of the algorithm features that we consider to be advantageous compared to previous works:

- Fast computation time due to simplified morphology-based techniques.
- Efficient in terms of the required memory and real time response.
- Invariant to color and size of the human hand.
- Robust against inherent hardware noise from currently available webcams.

https://doi.org/10.1017/S0263574714000083 Published online by Cambridge University Press

- Sharpness of the human hand image is not necessary; hence, the camera focus does not need to be constantly adjusted.
- Applicable to hands with or without gloves.
- Tested with constant color backgrounds or long depth of field in natural environment backgrounds.
- Can be integrated in a single system or on multi-systems over a network.
- Does not require camera calibrations.

References

1. V. I. Pavlovic, R. Sharma and T. S. Huang, "Visual interpretation of hand gesturesfor human-computer interaction: A review," *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 677–695 (1997).
2. T. H. H. Maung, "Real-Time hand tracking and gesturerecognition system using neural networks," *World Acad. Sci., Eng. Technol.* **38**, 470–474 (2009).
3. C. Manresa, J. Varona, R. Mas and F. J. Perales, "Hand tracking and gesture recognition forhuman-computer interaction," *Electron. Lett. Comput. Vis. Andimage Anal.* **5**, 96–104 (2005).
4. M. Elmezain, A. Al-Hamadi, R. Niese and B. Michaelis, "A robust method for hand tracking using mean-shift algorithm and kalman filter in stereo colour image sequences," *World Acad. Sci. Eng. Technol.* (35), 283–287 (Nov. 2009).
5. J. S. Cui and Z. Q. Su, "Vision-Based Hand Motion Capture Using Genetic Algorithm," Lecture Notes in Computer Science - Applications of Evolutionary Computing, vol. 3005, pp. 289–300, 2004.
6. N. D. Binh, E. Shuichi and T. Ejima, "Real-Time Hand Tracking and Gesture Recognition System," *Proceedings of International Conference on Graphics, Vision and Image* (Dec. 19–21, 2005) pp. 362–368.
7. H. K. Lee and J. H. Kim, "An HMM-Based threshold model approach for gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(10), 961–973 (Oct. 1999).
8. T. Starner and A. Pentland, "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models," *Technical Report 375*, MIT Media Lab, Perceptual Computing Group (1995).
9. R. Kjeldsen and J. Kender, "Visual Hand Gesture Recognition for Window System Control," *Proceedings of Int. Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland (1995) pp. 184–188.
10. M. Zhao, F. K. H. Quek and X. Wu, "RIEVL: Recursive induction learning in hand gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(11), 1174–1185 (Nov. 1998).
11. H. Yoon, J. Soh, Y. J. Baeand and H. S. Yang, "Hand gesture recognition using combined features of location, angle and velocity," *Pattern Recognit.* **34**(7), 1491–1501 (2001).
12. R. Lockton and A. W. Fitzgibbon, "Real-Time Gesture Recognition Using Deterministic Boosting," *Proceedings of the British Machine Vision Conference* (2002) pp. 817–826.
13. R. Bowden, D. Windridge, T. Kadir, A. Zisserman and M. Brady, "A linguistic Feature Vector for the Visual Interpretation of Sign Language," *Proceedings of the European Conference on Computer Vision* (2004) pp. 390–401.
14. M. Jeon, S. Wan Lee and Z. Bien, "Hand gesture recognition using multivariate Fuzzy Decision Tree and user adaptation," *Int. J. Fuzzy Syst. Appl.* **1**(3), 15–31 (2011).
15. G. Caridakis, K. Karpouzis, A. Drosopoulos and S. Kollias, "SOMM: Self organizing Markov map for gesture recognition," *Pattern Recognit. Lett.* **31**(1), 52–59 (2010).
16. H. Suk, B. Sin and S. Lee, "Hand gesture recognition based on dynamic Bayesian network framework," *Pattern Recognit.* **43**(9), 3059–3072 (Sep. 2010) Department of Brain and Cognitive Engineering, Korea University.
17. C. Tomasi, S. Petrov and A. Sastry, "3D Tracking = Classification + Interpolation," *Proceedings of the IEEE International Conference on Computer Vision*, **2**, (Oct. 2003) pp. 1441–1448.
18. L. Lam, S. W. Lee and C. Y. Suen, "Thinning methodologies- A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(9), 869–885 (Sep. 1992).