

Model-based 3D hand posture estimation from a single 2D image

Chin-Seng Chua*, Haiying Guan, Yeong-Khing Ho

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore 639798

Received 9 October 2000; received in revised form 6 October 2001; accepted 31 October 2001

Abstract

Passive sensing of the 3D geometric posture of the human hand has been studied extensively over the past decade. However, these research efforts have been hampered by the computational complexity caused by inverse kinematics and 3D reconstruction. In this paper, our objective focuses on 3D hand posture estimation based on a single 2D image. We introduce the human hand model with 27 degrees of freedom (DOFs) and analyze some of its constraints to reduce the 27 to 12 DOFs without any significant degradation of performance. A novel algorithm to estimate the 3D hand posture from eight 2D projected feature points is proposed. Experimental results using real images confirm that our algorithm gives good estimates of the 3D hand pose. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: 3D hand posture estimation; Model-based approach; Gesture recognition

1. Introduction

Hand posture analysis is an interesting research field and has received much attention in recent years. It is the pivotal context of particular applications such as gesture recognition [1–4], human computer interaction (HCI) [5], sign language recognition (SLR) [6–8], virtual reality (VR), computer graphic animation (CGA) and medical studies.

General solutions for posture analysis are divided into two categories: One attempt is to use mechanical devices, such as glove-based devices, to directly measure hand joint angles and spatial positions. The other attempt uses computer vision-based techniques. Although the former can give real-time processing and reliable information, it requires the user to wear a cumbersome device and generally carry a load of cables that connect the device to a computer. All these requirements make the sensing of natural hand motion difficult. On the other hand, the latter is suitable for hand posture estimation since vision is a non-invasive way of sensing.

Vision-based approaches can be classified into two types: appearance- and three-dimensional (3D) model-based approach. The appearance-based methods are mainly based on the visual image model and use the image templates to describe the postures. The gestures are modeled by relating the appearance of any gesture to the appearance of the set of predefined, template gestures. Starner et al. [6]

use silhouette moments as the features to analyze the American sign language (ASL). In their research project, “Real-time American Sign Language Recognition Using Desktop and Wearable Computer Based Video”, they present two real-time hidden Markov model-based systems for recognizing sentence-level continuous ASL using a single camera to track the user’s unadorned hands.

The major advantage of this approach is the simplicity of their parameter computation. However, the loss of precise spatial information makes them less suitable for manipulative hand posture analysis. Since appearance-based methods are sensitive to viewpoint changes and cannot provide precise spatial information, it is less suited for manipulative and interactive applications.

Conventional model-based methods are mainly used in two areas: 3D hand tracking and 3D hand posture estimation. Hand tracking is to locally track and estimate the positions of joints and tips of the hand in the image sequence. By analysis of static and dynamic motions of the human hand, Lee and Knuii [9] present some constraints on the joints and use them to simulate the human hand in real images. In the experiments, they used markers to identify the fingertips. On the basis of Lee’s contribution, Lien et al. [10] proposed a fast hand model fitting method for the tracking of hand motion. Although they improve the performance of the tracking algorithm, the computation of inverse kinematics is still required.

Rehg [11] described DigitalEyes for a real-time hand tracking system, in which the articulated motion of fingers was recognized as a 3D mouse by using a hand model

* Corresponding author. Tel.: +65-790-5412; fax: +65-793-3318.

E-mail address: ecschua@ntu.edu.sg (C.-S. Chua).

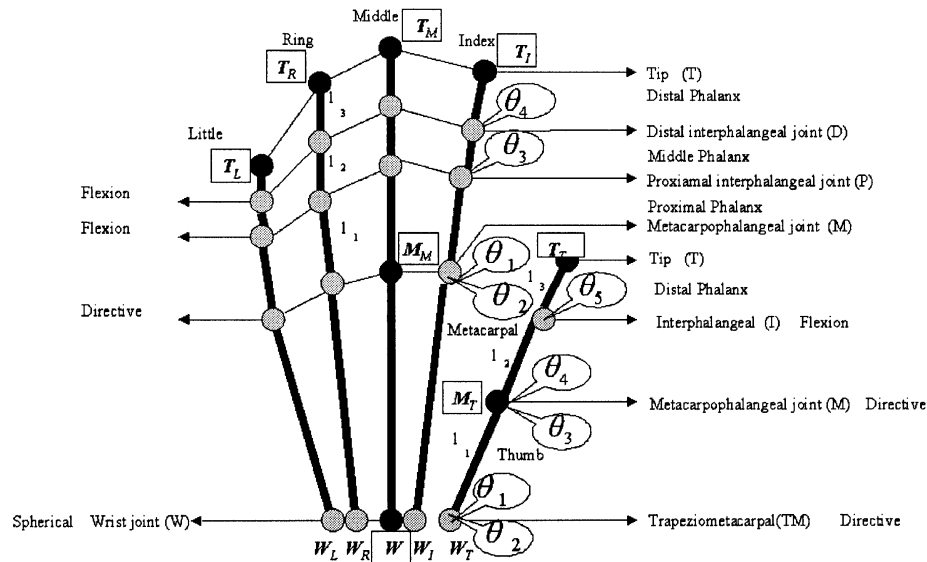


Fig. 1. Twenty-seven DOFs hand model.

having 27 degrees of freedoms (DOFs). This approach was based on the assumption that the positions of fingertips in the human hand, relative to the palm, are almost always sufficient to differentiate a finite number of different gestures. The hand gesture was estimated by a non-linear least squares method that minimizes the residual distances in finger links and tips of the model hand and those of the observed hand.

Shimada et al. [4] present a method to track the pose (joint angles) of a moving hand and refine the 3D shape (widths and lengths) of the given hand model from a monocular image sequence. First, the algorithm uses the silhouette features and motion prediction to obtain the approximated 3D shape. Then, with inequality constraints, they refine the estimation by the extended Kalman filter (EKF).

Without the motion information, some research efforts have concentrated on 3D hand posture estimation. In the study of Chang [12], a prototype system for estimating the position and orientation of a human hand as well as the joint angles of the thumb and the fingers from a single image is developed. The hand pose is estimated by using sparse range data generated by laser beams and by using the generalized Hough transform. Possible configurations for the fingers and the thumb are generated by the inverse kinematic technique.

Although the above algorithms have promising results, posture estimation is not yet advanced enough to provide a flexible and reliable performance for potential applications. The estimation of kinematic parameters from the detected features is a complex and cumbersome task. They face the following problems: First, the articulated mechanism of the human hand, which involves high DOF, is more difficult to analyze than a single rigid object: its state space is larger and its appearance is more complicated. Second, model-based methods always involve finding the

inverse kinematics, which are in general ill posed. It is obviously a task of computational complexity to estimate these kinematic parameters from the detected features. Third, previous methods on 3D require the use of multiple cameras, which not only is resource consuming, but also needs some form of 3D reconstruction that itself is computationally intense. Finally, it should be pointed out that the knowledge of exact hand posture parameters seems unnecessary for the recognition of communicative gestures.

In this paper, the goal of our work is to avoid the complex computation of inverse kinematics and 3D reconstruction; that is, without using 3D information, we propose a new approach to estimate the 3D hand posture from a single two-dimensional (2D) image. Preliminary results can be found in Refs. [13,14], which deals only with finger posture. This paper extends the idea further to compute the 3D posture for the entire hand. First, we analyze the human hand model with 27 DOFs and its constraints. The constraints play an important role in our study, which help us to reduce 27 to 12 DOFs without significant degradation of performance. Using the hand model and its constraints, we develop an algorithm to estimate the 3D hand posture by using eight feature points to retrieve the 3D hand posture. The eight feature points are the point of wrist, the tips of the fingers and thumb, and the metacarpophalangeal joints for the middle finger and thumb. We use color markers to identify these eight points and retrieve the approximate posture of the hand. Occlusion of any of the eight points is not considered in this paper.

In the experiments, two feature extraction methods are utilized: one for model building and the other for on-line hand posture estimation. In extracting the parameters for the hand model, a higher degree of accuracy in detecting the feature points is necessary. In this regard, the feature points are extracted from the silhouette contour of the

out-stretched hand. For on-line hand posture estimation, the silhouette contour may not contain essential feature points (say, the tips of the fingers of a clenched fist). Color markers, placed on the necessary positions of the hand are utilized. Pose estimation results obtained from real images are shown by comparison. These results confirm that our algorithm gives correct hand posture estimation.

This paper is organized as follows: Section 2 discusses the hand model and its constraints. Section 3 presents the methodology to estimate the hand posture. Two test cases involving various degrees of finger-extension are investigated in Section 4.

2. Hand model and its constraints

2.1. DOFs hand model

Lee and Knuii [9] defined a hand model with 27 DOFs. The joints of the human hand are classified into three kinds: flexion, directive or spherical joints, which consist of one DOFs (extension/flexion), two DOFs (one for extension/flexion and one for adduction/abduction) and three DOFs (rotation), respectively, (see Fig. 1). For each finger, there are four DOFs described by θ_1 – θ_4 . The thumb has five DOFs described by θ_1 – θ_5 . Including the six DOFs for the translation and rotation of the wrist, the model has 27 DOFs. Using the Denavit–Hartenberg (D–H) representation (briefly described in Appendix A), the forward kinematics of the finger links and thumb links are described in Appendices B and C, respectively. The reader may refer to Refs. [13,14] for more details.

2.2. Model constraints

Conventional models of the human hand are lacking in constraints. It limits their usefulness in computer vision and animation. The lack of constraints leads to unnatural model behavior. On the other hand, because the movements of the fingers are inter-dependent in the human hand and in order to reduce search space of matching, constraints are essential to further realize the hand motion.

2.2.1. Constraint 1

This constraint is proposed by Rijpkema [15]. The angles of *D* (Distal) joints and *P* (Proximal) joints are dependent

$$\theta_4 = \frac{2}{3}\theta_3 \quad (1)$$

where θ_3 and θ_4 represent the extension/flexion DOF for the *P* and *D* joints of each finger (see Fig. 1). This is a strong constraint and is widely adopted by many researchers working on hand analysis. From this constraint, the DOF of each finger can be reduced from 4 to 3 DOFs. Two of them are located at the *M* joints. The other one controls both *P* and *D* flexion joints.

2.2.2. Constraint 2

This constraint is proposed by Rijpkema [15]. The thumb is a more complicated manipulator, because a large part of the thumb is part of the palm and the joints are moving along non-trivial axes. However, it is known that the kinematics of the thumb can be calculated almost uniquely by experimental observations

$$\theta_1 = 2\left(\theta_3 - \frac{1}{6}\pi\right) \quad (2)$$

$$\theta_2 = \theta_4 \frac{7}{5}. \quad (3)$$

These two equations help us to reduce the DOF of the thumb from 5 to 3 DOFs. The first is for extension/flexion movements of the *TM* joint and the *M* joint, the second is for the adduction/abduction movements of the *TM* joint and the *M* joint, and the last is for the extension/flexion of the *I* joint (see Fig. 1 for the joint definition).

2.2.3. Constraint 3

This constraint is proposed by Kuch and Huang [16]. The joint angles of *P* and *M* joints of finger have a dependency represented by the following equation:

$$\theta_1 = k\theta_3 \quad 0 \leq k \leq 1/2 \quad (4)$$

where θ_1 and θ_3 represent extension/flexion DOF for the *M* and *P* joints of the finger, respectively. Kuch assumes that $k = 1/2$. This constraint is suitable for dynamic cases; in static analysis, we call this as a ‘weak constraint’. In our experiments, the feature extraction may some times be difficult (due to occlusion or juxtaposition of features) and this constraint may fail. If this constraint fails and leads to an invalid solution, the value of coefficient k is automatically adjusted between the range of 0–0.5 to give the next best approximate solution.

2.2.4. Constraint 4

This constraint is proposed by Lee and Kunii [17]. There is little adduction (add.)/abduction (abd.) of the *M* (metacarpophalangeal) joint of the middle finger; that is

$$\theta_2 = 0 \quad (5)$$

where θ_2 is adduction/abduction DOF for the *M* joint of the middle finger.

According to the hand model we have established, we propose the following constraints.

2.2.5. Constraint 5

Five points (Wrist joint, Metacarpalangeal joint, Proximal joint, Distal joint and Tip represented by *W*, *M*, *P*, *D* and *T*, respectively) of each finger are coplanar. We define this plane as the ‘finger plane’. According to Constraint 4, we omit the adduction/abduction DOF for the *M* joint, so that *M*, *P*, *D* joints of four fingers are all extension/flexion joints.

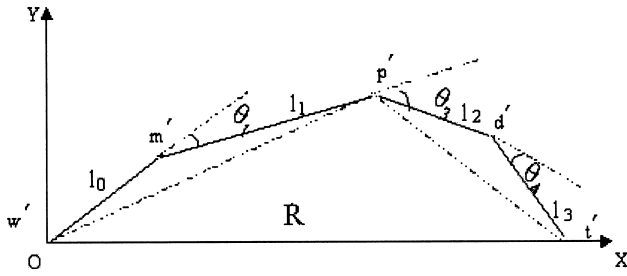


Fig. 2. Pentagon geometry.

For accuracy, we may define a different W point for each finger (see Fig. 1).

2.2.6. Constraint 6

The joint angles of I and M of the thumb have a dependency represented by the following equation:

$$\theta_5 = a\theta_4 \quad a \geq 0 \quad (6)$$

This is also a ‘weak constraint’. We use this to estimate the thumb posture.

2.2.7. Constraint 7

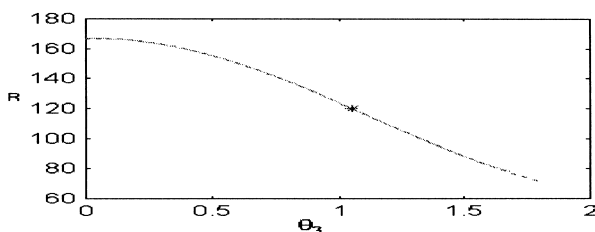
Four points (Trapeziometacarpal joint, Metacarpalangeal joint, Interpalangeal joint and thumb tip represented by TM , M , I and T , respectively) of the thumb are coplanar. We define this plane as the ‘thumb plane’. In the experiments, the TM joint is approximated by the W_T point (see Fig. 1).

2.2.8. Constraint 8

The palm is assumed not to become hollow in a non-prehensile configuration. We assume that the M joint of each finger are located in the plane called ‘palm plane’ and this plane is perpendicular to the ‘finger plane’ of the middle finger.

In general, three points are necessary to align the palm with the images. However, if we assume that the palm does not become hollow, it is impossible to move the palm when the ‘middle finger plane’ is fixed.

Using Constraint 1, 3–5, we omit three DOFs for each finger. Using Constraint 2 and 6, we omit three DOFs for the thumb. Over all, our simplified model is reduced from 27 to 12 DOFs.

Fig. 3. Regression curve for θ_3 .

3. Hand posture estimation

3.1. Problem description

The purpose of this part is to analyze the geometric characteristics of the hand and replicate its pose based on 2D projected information. Without loss of generality, we assume that the world coordinate frame is aligned with the camera frame; in other words, the image plane coincides with the X – Y frame of the world. In our experiments, we adopt color markers to identify the feature points of the joints and tips.

3.2. Finger posture estimation

3.2.1. Solution for five points in the 2D ‘finger plane’

For a particular finger, we define the 3D distance between the tip of finger T to the wrist joint W as R . We denote the positions of W , M , P , D and T in the ‘finger plane’¹ by $w'(0, 0)$, $m'(X'_M, Y'_M)$, $p'(X'_P, Y'_P)$, $d'(X'_D, Y'_D)$ and $t'(X'_T, Y'_T)$. The coordinate system is oriented as shown in Fig. 2. Their 3D coordinates in this plane are $W'(0, 0, 0)$, $M'(X'_M, Y'_M, 0)$, $P'(X'_P, Y'_P, 0)$, $D'(X'_D, Y'_D, 0)$ and $T'(X'_T, Y'_T, 0)$. In the pentagon (see Fig. 2), which consists of the five points W , M , P , D and T , the lengths of l_0 , l_1 , l_2 and l_3 are constant parameters. From the pentagon, we obtain the following equations:

$$(w'p')^2 = l_0^2 + l_1^2 - 2l_0l_1 \cos(\pi - \theta_1) \quad (7)$$

$$(p't')^2 = l_2^2 + l_3^2 - 2l_2l_3 \cos(\pi - \theta_4) \quad (8)$$

$$\cos \angle m'p'w' = \frac{l_1^2 + (w'p')^2 - l_0^2}{2l_1(w'p')} \quad (9)$$

$$\cos \angle d'p't' = \frac{l_2^2 + (p't')^2 - l_3^2}{2l_2(p't')} \quad (10)$$

$$\angle w'p't' = \pi - \theta_3 - \angle m'p'w' - \angle d'p't' \quad (11)$$

$$R^2 = (w'p')^2 + (p't')^2 - 2(w'p')(p't')\cos \angle w'p't' \quad (12)$$

For a given θ_3 , the values of θ_1 and θ_4 are calculated by Eqs. (1) and (4). Using Eqs. (B5) and (B6), we obtain the values of $w'p'$ and $p't'$. Substituting these values into Eqs. (C1) and (C2), the angles $\angle m'p'w'$ and $\angle d'p't'$ are obtained, from which $\angle w'p't'$ can be calculated (Eq. (C3)). Finally, the corresponding R for the given θ_3 is obtained (Eq. (C4)). The relationship between θ_3 and R are shown in Fig. 3.

¹ Positions with $'$ are used to denote oriented finger plane positions while those without $'$, which will be introduced in Section 3.2.2, are used to denote positions referenced from the image plane.

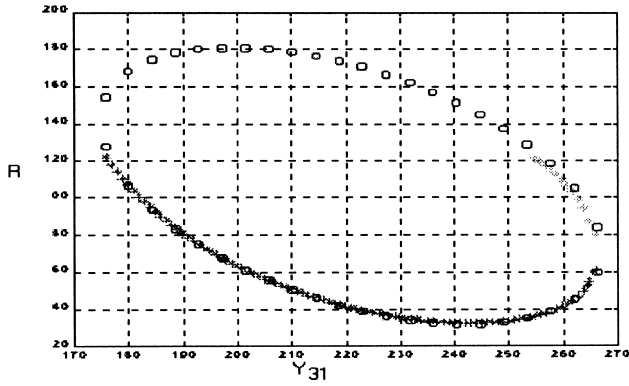


Fig. 4. The relationship between the R and Y_{31} .

From Fig. 3, the curve can be approximated by a cubic polynomial. Using a regression method, for a given R , we can find the shape of the pentagon. Hence, we can conclude that if the distance R between the W (W joint) and T (Tip) is given, the shape of the pentagon is determined. The positions of W' , M' , P' , D' and T' are obtained at the same time.

3.2.2. 3D solution for the five points

We define the 3D coordinates of the wrist points and finger points (in the camera frame) as $W(X_W, Y_W, Z_W)$, $M(X_M, Y_M, Z_M)$, $P(X_P, Y_P, Z_P)$, $D(X_D, Y_D, Z_D)$ and $T(X_T, Y_T, Z_T)$. Their orthographic projection coordinates in the image plane are: $w(X_W, Y_W)$, $m(X_M, Y_M)$, $p(X_P, Y_P)$, $d(X_D, Y_D)$ and $t(X_T, Y_T)$.

In Section 3.2.1, it was shown that for a given value of R and link constants (l_0 , l_1 , l_2 and l_3), θ_3 and hence the positions of W' , M' , P' , D' and T' in the finger plane can be determined. The task at hand is to find the 3D positions (relative to the camera frame) of the finger joints and tip (that is, W , M , P , D and T) given any three 2D image positions of these points (that is, w , m , p , d or t). For ease of description, we define the 3D coordinates of the detected finger points as $P_1(X_1, Y_1, Z_1)$, $P_2(X_2, Y_2, Z_2)$, and $P_3(X_3, Y_3, Z_3)$ which are any three points of W , M , P , D and T . Their orthographic projection coordinates in the image plane are: $p_1(X_1, Y_1)$, $p_2(X_2, Y_2)$, and $p_3(X_3, Y_3)$. Taking P_1 as the reference, the positions of P_2 and P_3 are defined by $P_{21}(X_{21}, Y_{21}, Z_{21})$ and $P_{31}(X_{31}, Y_{31}, Z_{31})$, where $X_{21} = X_2 - X_1$, $Y_{21} = Y_2 - Y_1$, $X_{31} = X_3 - X_1$, and $Y_{31} = Y_3 - Y_1$. Rotating them to the finger plane, their new 2D coordinates are p'_1 , p'_2 , p'_3 and 3D coordinates are P'_1 , P'_2 , P'_3 , respectively.

In order to find the solution for this problem, the following steps are proposed:

- Obtain the 3D hand model parameters (the lengths of the finger links or l_1 , l_2 and l_3) from the 2D image information by using the feature extraction method (see Section 4.2 for details).

- For simplicity of computation, we treat Y_{31} as a variable and R as known.² For each possible value of R , we calculate the positions of w' , m' , p' , d' , t' in 2D plane by using the algorithm described in Section 3.2.1. We obtain the distances among P_1 , P_2 and P_3 and let them be l_{21} , l_{23} and l_{31} (the distance between P_2P_1 , P_2P_3 and P_3P_1 , respectively). Z_{21} is described by the following equation:

$$Z_{21} = \sqrt{(l_{21}^2 - X_{21}^2 - Y_{21}^2)} \quad (13)$$

- Obtain Y_{31} and Z_{31} using the following equations:

$$X_{31}^2 + Y_{31}^2 + Z_{31}^2 = l_{31}^2 \quad (14)$$

$$(X_{31} - X_{21})^2 + (Y_{31} - Y_{21})^2 + (Z_{31} - Z_{21})^2 = l_{23}^2 \quad (15)$$

From the above equations, X_{21} , Y_{21} , Z_{21} , X_{31} , l_{31} and l_{23} are known. Two sets of solutions for Y_{31} and Z_{31} are obtained. The following constraints are used to choose the correct one: since the finger tips must flex towards the palm, the hand is defined in a particular orientation; that is, if the palm orientation is given, one set of solutions is omitted.

- Calculating all possible values of R , we obtain the relationship between Y_{31} and R (Fig. 4).
- It is shown that the relationship between Y_{31} and R can be approximated by the quadratic curve (Fig. 4). Using the regression method, the curve function can be approximated. For the actual Y_{31} , we obtain R .
- Using the computed R , we find the 2D solutions of w' , m' , p' , d' and t' with the method described in Section 3.2.1. The equivalent 3D coordinates of W' , M' , P' , D' and T' in the 'finger plane' are extracted.
- Compute the rotation matrix mapping points P'_1 , P'_2 and P'_3 to points P_1 , P_2 and P_3 .
- Obtain the 3D position of the other points with the rotation matrix. For correctness of solution, the five points, W , M , P , D and T should be coplanar.
- Earlier, we assume $k = 1/2$ (Eq. (4)). If there are no solutions, it could be caused by the deviation of this weak constraint. We vary the value of the coefficient, k , obtain the relationship between the θ_3 and R and recalculate the 3D positions until we find an approximate solution. Fig. 10(a) shows the posture in which the value of k is near the extreme of zero. In our experiments,

² Since $Y_{31} = Y_3 - Y_1$ is the difference between y-ordinates of image points, one may argue that Y_{31} is already known and should not be considered a variable. However, in doing so, one will have to contend with l_{21} , l_{31} and l_{23} of Eqs. (13)–(15). A simple approach would be to treat Y_{31} and Z_{31} as variables and l_{21} , l_{31} and l_{23} as functions of a given R ; that is, if R is known, l_{21} , l_{31} and l_{23} can be computed as in the previous section. Hence, Eqs. (14) and (15) will contain the two variables of Y_{31} and $sf1_{31}$ since the other parameters (X_{21} , Y_{21} , Z_{21} , X_{31} , l_{31} and l_{23}) can be calculated. Using the two equations, Y_{31} is solved for that given $sf1$ value. By simulating different values of R , the solutions for Y_{31} can be computed. The relationship between Y_{31} and $sf1$ can be approximated by quadratic curve (Fig. 4) from which the actual R value can be identified (since Y_{31} is actually known).



Fig. 5. Initial frame.

the value of k is automatically varied from 0.5 to 0 in variable steps ($k = (1/2)^t$ (t is the number of the steps)) until a solution is obtained (in normal case, the solution can be obtained within one or two steps).

3.3. Thumb posture estimation

Combined with the Constraint 4 (Eq. (3)) and Constraint 5 (Eq. (6)), we use a similar approach as above to obtain the four-point solution for the thumb. In the algorithm, we use the three 2D feature points of the thumb (that is, TM, MP and T) to calculate the 3D positions of the four points: TM, MP, IP and T (see Fig. 1).

3.4. Hand posture estimation

On the basis of the algorithms for finger posture estimation and thumb posture estimation which is described in Sections 3.2 and 3.3. We present a hand posture estimation algorithm by using eight feature points of the hand; that is, the wrist point (W), the metacarpal joint of the middle finger (M_M), the metacarpophalangeal joint of the thumb (M_T), the four tips of the fingers (T_I, T_M, T_R and T_L) and the tip of the thumb (T_T).

From the feature extraction stage, we obtain the W point of the middle finger. For the 2D positions of the three feature points of middle finger (that is, W, M_M and T_M), we use the

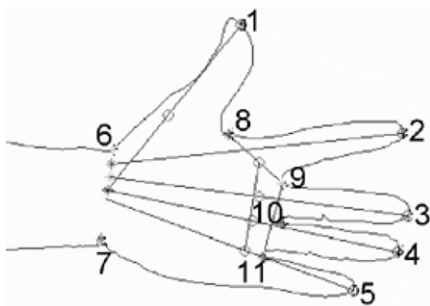


Fig. 6. Contour.

Table 1
Model parameters of the hand (in pixels)

	Thumb	Index finger	Middle finger	Ring finger	Little finger
l_0	–	120	121	118	117
d	–	–27	0	19	43
l_1	78	59	62	61	43
l_2	63	30	33	31	24
l_3	51	29	26	27	26

finger posture solution algorithm (Section 3.2) to retrieve the middle finger posture.

Once the 3D posture of the middle finger is determined, the ‘finger plane’ for the middle finger in 3D space is known. According to Constraint 8 (see Section 2.2, the ‘palm plane’ is perpendicular to the ‘finger plane’ of the middle finger. This constraint, together with the knowledge of two points (M_M and W joints) on the palm, constrains the palm plane. From this plane, and using the predetermined hand model, the positions of the metacarpal (M) joint of the index, ring and little finger can be identified. To obtain better finger plane estimations for the other fingers (index, ring and little finger), different wrist points for each finger (W_I, W_R and W_L) are approximated using the hand model.

For the three feature points (W, M and T) of each of index, ring and little fingers, we use the finger posture algorithm (see Section 3.2 for details) to retrieve the 3D posture. In the same way, for the three feature points (W_T, M_T and T_T)



Fig. 7. Original image.

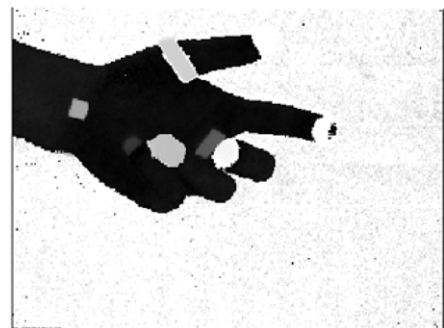


Fig. 8. Hue image.

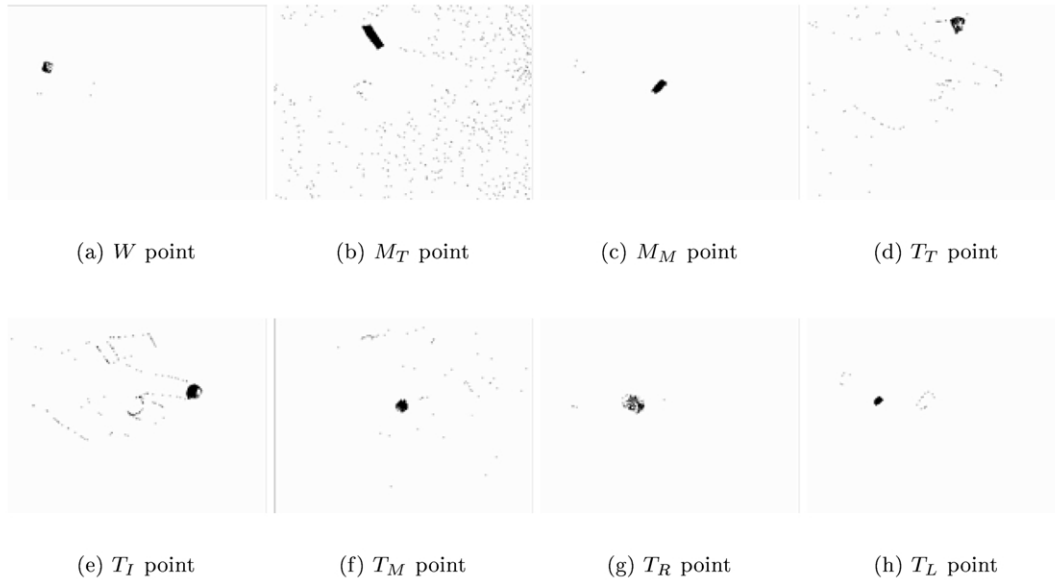


Fig. 9. Points detection by color markers: (a) W point; (b) M_T point; (c) M_M point; (d) T_T point; (e) T_I point; (f) T_M point; (g) T_R point; (h) T_L point.

of the thumb, the thumb posture is retrieved by the thumb posture algorithm. Finally, we synthesize the hand posture in 3D space.

4. Experiments

4.1. Model parameter estimation

4.1.1. Feature detection based on hand contour in initial frame

Feature detection/extraction stage is concerned with the detection of features which is used for the estimation of the parameters of the chosen hand model. It affects the accuracy of the model parameters and so does the estimation results. For the accuracy, we use the following steps to detect the feature with hand contour:

1. From the original image (see Fig. 5), extract the contour of the open hand using the LOG operator [18].
2. Calculate the contour curvature [19].
3. Obtain the local maximum and minimum curvature points as feature points (see Fig. 6).

4.1.2. Model parameter estimation

Referring to Fig. 6, the five minimum curvature points are the five convex points (numbered by 1–5) of the hand contour; that is, the tips of the fingers or thumb. The first four maximum curvature points are the concave points of the hand contour (numbered by 8–11). The fifth and sixth maximum curvature points (numbered by 6 and 7) are the feature points for the wrist. Using the two detected feature points of the wrist, we estimate the distances of the wrist points (W_L , W_R ,

W , W_I , W_T) for all the fingers by ratio. The finger link lengths (l_0 – l_3 for each finger and l_1 – l_3 for the thumb) are calculated according to assumed ratios between finger segments. With the detected features, we obtain the model parameters (shown in Fig. 6) and tabulated in Table 1.

4.2. Feature detection by color markers

For simplicity of feature extraction, color markers are commonly used by other research groups to identify the main feature points. We use color markers to detect the eight main points in the real image. Eight points (W , M_T , M_M , T_T , T_I , T_M , T_R and T_L) are identified from the centroids of the detected color regions (see Fig. 1 for definition).³ Figs. 7 and 8 show the initial frame and its hue image. Fig. 9 shows the classification results.

4.3. Hand posture estimation using several feature points

Two test cases are presented. The first test case (Fig. 10) shows hand postures with equal degrees of extension for each finger and varying from a slightly clenched configuration to an almost fully clenched configuration. The second test case (Fig. 11) shows hand postures with different degrees of extension for each finger. For each posture, we show the original images (left image), the estimated 3D pose overlaid on the original image (center image) and the 3D pose from a different viewpoint (right image). The link angles for the fingers and thumb of both test cases are tabulated in Table 2. It is interesting to note that Constraints 3 is violated for the posture in Fig. 10(a) with the assumption of $k = 1/2$. For Fig. 10(a) with the

³ The subscripts represent different fingers and thumb.

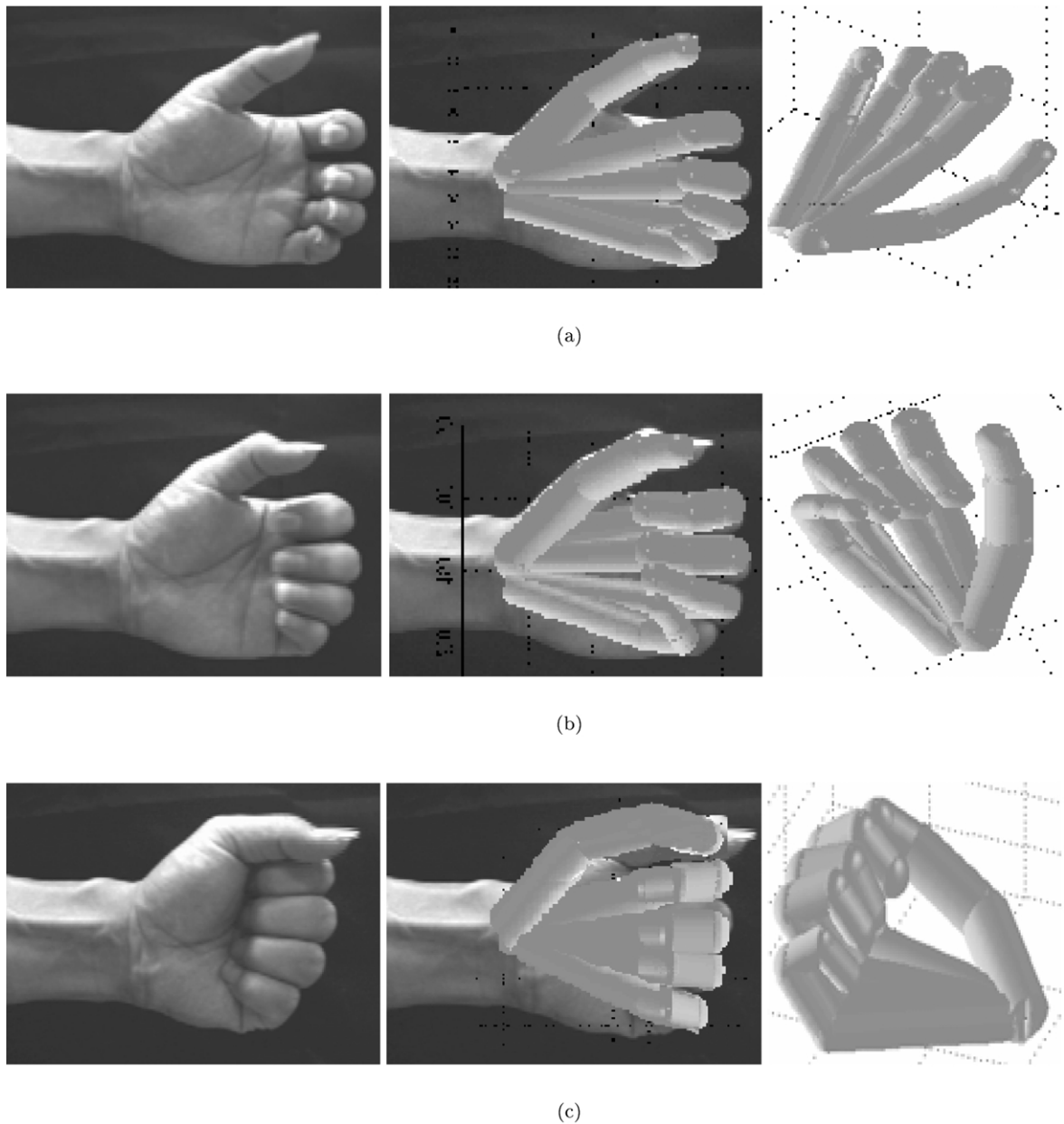
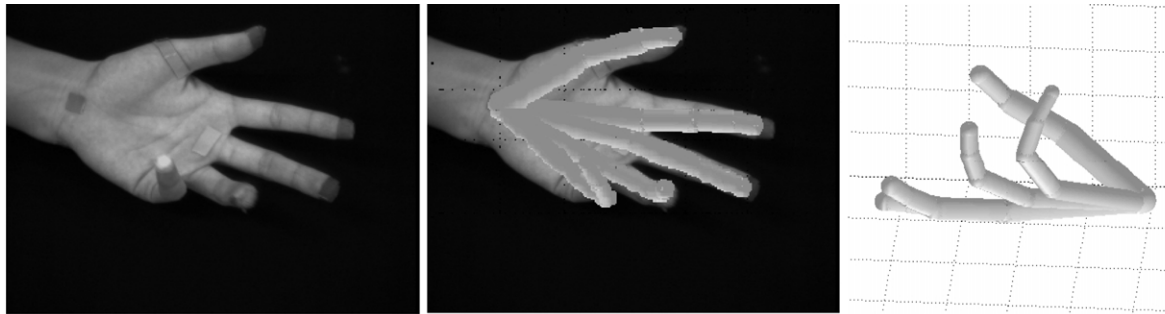


Fig. 10. Hand postures in clenching motion: (a) slightly-clenched configuration; (b) moderately-clenched configuration and (c) almost fully clenched configuration. Left: Original image; Center: Estimated 3D pose overlaid on image; Right: 3D pose from a different viewpoint.

extended fingers, θ_1 is almost zero and hence $k \cong 0$. In this case, the algorithm is able to simulate for varying values of k until a solution is obtained. In our experiments, k is varied from 0.5 to 0 ($k = (1/2)^t$ (t is the number of the steps)) until a solution is obtained. Also, we show in Fig. 11(c) that the posture for non-frontal views of the hand can be retrieved as well. The constraint that we place on our experiments is that all eight color markers (identifying the eight positions of W_T , M_T , M_M , T_T , T_b , T_M , T_R , T_L) be visible. Occlusion of one or several markers is out of the scope of this paper and is currently under investigation.

5. Conclusion

In this paper, we proposed an algorithm to estimate the 3D hand posture using a single 2D image. The new algorithm is promising because of the following reasons: First, the algorithm uses 2D positions of the feature points and avoids the computational complexity caused by the 3D reconstruction. Second, the algorithm does not involve the computation of inverse kinematics. Third, the algorithm uses only single 2D image to retrieve the 3D hand posture. There is no need to know the motion information of fingers or thumb that can only be provided by a sequence of images.



(a)



(b)



(c)

Fig. 11. Hand postures with different degrees of extension for each finger (using color markers): (a) slightly-clenched configuration; (b) moderately clenched configuration and (c) moderately clenched configuration from a non-frontal viewpoint. Left: original image; Center: estimated 3D pose overlaid on image; Right: 3D pose from a different viewpoint.

Table 2

Link angles for fingers and thumb. The values with * are not satisfied with $k = 1/2$. Additional iterations are required to compute the valid solution

		Thumb		Index		Middle		Ring		Little	
		θ_4	θ_5	θ_1	θ_3	θ_1	θ_3	θ_1	θ_3	θ_1	θ_3
Test case 1 (Fig. 12)	(a)	21.7	10.9	4.3*	67.0	7.1*	85.3	9.7*	78.2	14.3*	56.1
	(b)	19.2	9.6	30.5*	121.8	30.7*	122.7	27.2*	109.1	30.4	60.8
	(c)	20.6	10.3	50.7	101.4	54.5	109.0	52.6	105.2	51.4	102.8
Test case 2 (Fig. 13)	(a)	22.7	11.4	9.3	18.6	12.1	24.2	12.1	24.2	23.1	46.3
	(b)	25.7	12.8	12.2	24.4	30.3	60.6	26.4	52.8	34.7	69.4
	(c)	33.4	16.67	4.1	8.3	22.8	45.7	36.4	72.8	32.4	64.7

Experimental results have shown that our 3D hand posture estimation works well even with real images.

Appendix A. Denavit–Hartenberg representation

D–H representation is a commonly used convention for selecting frames of reference in robotic applications [20]. Using this convention, each homogeneous transformation matrix T_i^{i-1} is represented as a product of four ‘basic’ transformations with four parameters

$$T_i^{i-1} = \text{Rot}_Z(\theta_i)\text{Trans}_Z(d_i)\text{Trans}_X(a_i)\text{Rot}_X(\alpha_i) \tag{A1}$$

where T_i^{i-1} is homogeneous transform matrix from reference frame i to $i - 1$. $\text{Rot}_k(\text{angle}_i)$ is the rotation of angle $_i$ about the k axis and $\text{Trans}_k(\text{distance}_i)$ is the translation of distance $_i$ along the k axis.

According to the D–H rules, the local coordinate frames for each DOF of finger and thumb are defined. At the same time, the four D–H parameters which determine the transformation matrix between the two adjacent coordinates are obtained.

Appendix B. Forward kinematics of finger

From a kinematic point of view, the hand consists of multi-branched kinematic chains attached to a base. From our 27 DOFs hand model, each of the four fingers of the hand is a planar mechanism with four DOF, two of them at the M or Metacarpophalangeal joint (refer to Fig. 1), one at the P or Proximal Interphalangeal joint and one at the D or Distal Interphalangeal joint.

Using the D–H representation, the local coordinate system for each DOF of the joint and the parameters of the D–H representations are shown as Fig. 12. The 3D

model of the finger can be viewed as a set of six serial kinematic chains (finger links). All are attached to a base frame defined at the end of the palm (the frame 0). The D–H transformation matrices between the coordinate frames for one finger are described as

$$T_1^0 = \text{Rot}_Z\left(\frac{\pi}{2}\right)\text{Trans}_X(a_0)\text{Rot}_X\left(-\frac{\pi}{2}\right) \tag{B1}$$

$$T_2^1 = \text{Trans}_X(d_1)\text{Rot}_X\left(\frac{\pi}{2}\right) \tag{B2}$$

$$T_3^2 = \text{Rot}_Z(\theta_1)\text{Rot}_X\left(\frac{\pi}{2}\right) \tag{B3}$$

$$T_4^3 = \text{Rot}_Z(\theta_2)\text{Trans}_X(l_1) \tag{B4}$$

$$T_5^4 = \text{Rot}_Z(\theta_3)\text{Trans}_X(l_2) \tag{B5}$$

$$T_6^5 = \text{Rot}_Z(\theta_4)\text{Trans}_X(l_3) \tag{B6}$$

where l_1, l_2 and l_3 are the lengths of the finger links (Fig. 12). a_0 and d_1 are model parameters and are invariant for a given hand. $\theta_1, \theta_2, \theta_3$ and θ_4 are the four DOF of each finger, these act as variables to control the movement of each finger, relative to the palm’s orientation.

Appendix C. Forward kinematics of thumb

The thumb is very dexterous and therefore a more complicated manipulator. Since a large part of the thumb seems to be part of the palm, the motion of a thumb is not easily understood. Five DOFs are used to establish the link coordinate frames of the thumb: two of them at the TM or Trapeziometacarpal joint, two of them at the M or

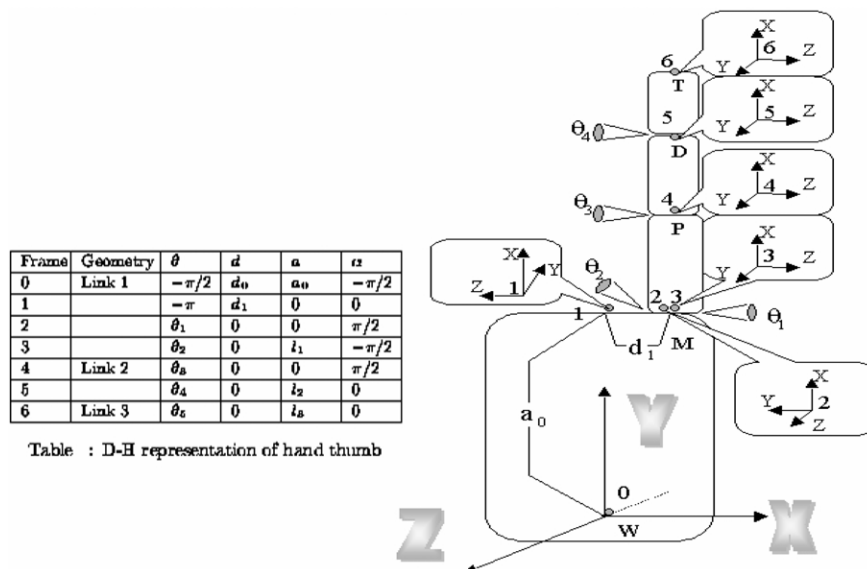


Fig. 12. Finger model and its D–H representations.

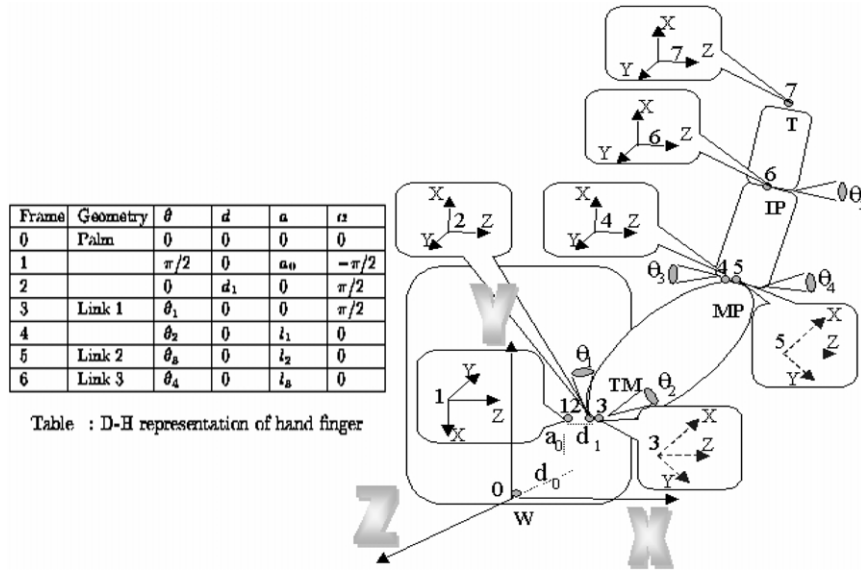


Fig. 13. Thumb model and its D–H representations.

Metacarpophalangeal joint, and one at the IP or Interphalangeal joint. The parameters of the D–H representation are shown in (refer to Fig. 13).

The transformation between each of the local coordinate frames of the thumb are described as

$$T_1^0 = \text{Rot}_Z\left(-\frac{\pi}{2}\right)\text{Trans}_Z(d_0)\text{Trans}_X(a_0)\text{Rot}_X\left(-\frac{\pi}{2}\right) \quad (\text{C1})$$

$$T_2^1 = \text{Rot}_Z(-\pi)\text{Trans}_Z(d_1) \quad (\text{C2})$$

$$T_3^2 = \text{Rot}_Z(\theta_1)\text{Rot}_X\left(\frac{\pi}{2}\right) \quad (\text{C3})$$

$$T_4^3 = \text{Rot}_Z(\theta_2)\text{Trans}_X(l_1)\text{Rot}_X\left(-\frac{\pi}{2}\right) \quad (\text{C4})$$

$$T_5^4 = \text{Rot}_Z(\theta_3)\text{Rot}_X\left(\frac{\pi}{2}\right) \quad (\text{C5})$$

$$T_6^5 = \text{Rot}_Z(\theta_4)\text{Trans}_X(l_2) \quad (\text{C6})$$

$$T_7^6 = \text{Rot}_Z(\theta_5)\text{Trans}_X(l_3) \quad (\text{C7})$$

where l_1 , l_2 and l_3 are the lengths of the thumb links (see Fig. 13). a_0 , d_0 and d_1 are model parameters and are invariant for a given hand. θ_1 , θ_2 , θ_3 , θ_4 and θ_5 are the five DOF of the thumb and act as variables to control the movement of the thumb.

References

- [1] J. Triesch, C. Von der Malsburg, A gesture interface for human–robot-interaction, The Proceedings of FG'98, Nara, Japan, 1998.
- [2] A.F. Bobick, A.D. Wilson, A state-based approach to the representation and recognition of gesture, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (12) (1997) 1325–1337.
- [3] T.J. Darrel, A. Pentland, Attention-driven expression and gesture analysis in an interactive environment, Proceedings of the First International Conference on Automatic Face and Gesture Recognition, 1995.
- [4] N. Shimada, Y. Shirai, J. Miura, Hand gesture estimation and model refinement using monocular camera—ambiguity limitation by inequality constraints, The Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, IEEE computer Society Press, Los Alamitos, CA, 1998, pp. 268–273.
- [5] V.I. Pavlovic, R. Sharma, T.S. Huang, Visual interpretation of hand gestures for human–computer interaction: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 677–695.
- [6] T.E. Starner, Unencumbered virtual environments, PhD Thesis, MIT Media Arts and Sciences Section, USA, 1993.
- [7] J.S. Kim, W. Jang, Z. Bien, A dynamic gesture recognition system for the korean sign language (ksl), IEEE Transactions on System, Man, and Cybernetics 26 (2) (1996) 354–359.
- [8] S. Tamura, S. Kawasaki, Recognition of sign language motion images, Pattern Recognition 21 (4) (1988) 343–353.
- [9] J. Lee, T.L. Knuii, Model-based analysis of hand posture, IEEE Computer Graphics and Application September (1995) 77–86.
- [10] C.C. Lien, C.L. Huang, Model-based articulated hand motion tracking for gesture recognition, Image and Vision Computing 16 (1998) 121–134.
- [11] J.M. Rehg, T. Kanade, Visual tracking of high DOF articulated structures: an application to human hand tracking, ECCV94 (1994) B35–B46.
- [12] C.C. Chang, W.H. Tsai, Model-based analysis of hand gestures from single images without using marked gloves or attaching marks on hands, Proceedings of the Fourth Asian Conference on Computer Vision, 2000, pp. 923–930.
- [13] C.S. Chua, H.Y. Guan, Y.K. Ho, Model-based finger posture estimation, Fourth Asian Conference on Computer Vision, Taipei, Taiwan, 2000, pp. 43–48.
- [14] H.Y. Guan, C.S. Chua, Y.K. Ho, Hand posture estimation from 2D monocular image, Second International Conference on 3-D Digital Imaging and Modeling, Ottawa, Canada, 1999, pp. 424–429.
- [15] H. Rijkema, M. Girard, Computer animation of knowledge-based

- human grasping, *IEEE Computer Graphics and Application* 25 (4) (1991) 339–348.
- [16] J.M. Kuch, T.S. Huang, Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration, *ICCV95* (1995) 666–671.
- [17] J. Lee, T.L. Kunii, Constraint-based hand animation, *Models and Techniques in Computer Animation* (1993) 110–127.
- [18] A. Huertas, G. Medioni, Detection of intensity changes with subpixel accuracy using Laplacian–Gaussian masks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (5) (1986) 651–664.
- [19] F. Mokhtarian, A. Mackworth, Scale-based description and recognition of planar curves and two-dimensional shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (1) (1986) 34–43.
- [20] M.W. Spong, *Robot Dynamics and Control*, Wiley, New York, 1989 chapter 3.