Appears in *Third European Conf. on Computer Vision*, Stockholm, Sweden, May 1994, pages 35-46.

Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking

James M. Rehg[1] and Takeo Kanade[2]

[1] Carnegie Mellon University, Department of Electrical and Computer Engineering, Pittsburgh PA 15213, jimr@cs.cmu.edu
[2] Carnegie Mellon University, The Robotics Institute, Pittsburgh PA 15213, tk@cs.cmu.edu

Abstract. Passive sensing of human hand and limb motion is important for a wide range of applications from human-computer interaction to athletic performance measurement. High degree of freedom articulated mechanisms like the human hand are difficult to track because of their large state space and complex image appearance. This article describes a model-based hand tracking system, called DigitEyes, that can recover the state of a 27 DOF hand model from ordinary gray scale images at speeds of up to 10 Hz.

1 Introduction

Sensing of human hand and limb motion is important in applications from Human-Computer Interaction (HCI) to athletic performance measurement. Current commercially available solutions are invasive, and require the user to don gloves [15] or wear targets [8]. This paper describes a noninvasive visual hand tracking system, called DigitEyes. We have demonstrated hand tracking at speeds of up to 10 Hz using line and point features extracted from gray scale images of unadorned, unmarked hands.

Most previous real-time visual 3D tracking work has addressed objects with 6 or 7 spatial degrees of freedom (DOF)[5, 7]. We present tracking results for branched kinematic chains with as many as 27 DOF (in the case of a human hand model). We show that simple, useful features can be extracted from natural images of the human hand. While difficult problems still remain in tracking through occlusions and across complicated backgrounds, these results demonstrate the potential of vision-based human motion sensing.

This paper has two parts. First, we describe the 3D visual tracking problem for objects with kinematic chains. Second, we show experimental results of tracking a 27 DOF hand model using two cameras.

2 The Articulated Mechanism Tracking Problem

Visual tracking is a sequential estimation problem: given an image sequence, recover the time-varying state of the world [5, 7, 14]. The solution has three basic

components: state model, feature measurement, and state estimation. The state model specifies a mapping from a state space, which characterizes all possible spatial configurations of the mechanism, to a feature space. For the hand, the state space encodes the pose of the palm (seven states for quaternion rotation and translation) and the joint angles of the fingers (four states per finger, five for the thumb), and is mapped to a set of image lines and points by the state model. A state estimate is calculated for each image by inverting the model to obtain the state vector that best fits the measured features. Features for the unmarked hand consist of finger link and tip occluding edges, which are extracted by local image operators.

Articulated mechanisms are more difficult to track than a single rigid object for two reasons: their state space is larger and their appearance is more complicated. First, the state space must represent additional kinematic DOFs not present in the single-object case, and the resulting estimation problem is more expensive computationally. In addition, kinematic singularities are introduced that are not present in the six DOF case. Singularities arise when a small change in a given state has no effect on the image features. They are currently dealt with by stabilizing the estimation algorithm. Second, high DOF mechanisms produce complex image patterns as their DOFs are exercised. People exploit this observation in making shapes from shadows cast by their hands.

To reduce the complexity of the hand motion, we employ a high image acquisition rate (10-15 Hz depending on the model) which limits the change in the hand state, and therefore image feature location, between frames. As a result, state estimation and feature measurement are *local*, rather than global, search problems. In the state space, we exploit this locality by linearizing the nonlinear state model around the previous estimate. The resulting linear estimation problem produces state corrections which are integrated over time to yield an estimated state trajectory. In the image, the projection of the previous estimate through the state model yields coordinate frames for feature extraction. We currently assume that the closest available feature is the correct match, which limits our system to scenes without occlusions or complicated backgrounds.

Previous work on tracking general articulated objects includes [14, 10, 9]. In [14], Yamamoto and Koshikawa describe a system for human body tracking using kinematic and geometric models. They give an example of tracking a single human arm and torso using optical flow features. Pentland and Horowitz [10] give an example of tracking the motion of a human figure using optical flow and an articulated deformable model. A much earlier system by O'Rourke and Badler [9] analyzed human body motion using constraint propagation.

In addition to the work on general articulated object tracking, several authors have developed specialized techniques for visual human motion analysis. This previous work differs from ours in two ways. First, markers or gloves are often used to simplify motion analysis [4]. Second, analysis is typically restricted to a subset of the total hand motion, such as a set of gestures [2] or rigid motion of the palm [1]. In [4], Dorner describes a system for interpreting American Sign Language from image sequences of a single hand. Dorner's system uses the full

set of the hand's DOFs, and employs a glove with colored markers to simplify feature extraction. Darrell and Pentland describe a system for learning and recognizing dynamic hand gestures in [2]. Their approach avoids the problems of hand modeling, but doesn't address 3D tracking. In other hand-specific work, Kang and Ikeuchi describe a range sensor-based approach to hand pose estimation [6], used in their Assembly Plan from Observation system. See [11] for a more extensive bibliography.

In order to apply the DigitEyes system to specific applications, such as HCI, two practical requirements must be met. First, the kinematics and geometry of the target hand must be known in advance, so that a state model can be constructed. Second, before local hand tracking can begin, the initial configuration of the hand must be known. We achieve this in practice by requiring the subject to place their hand in a certain pose and location to initiate tracking. A 3D mouse interface based on visual hand tracking is presented in [11].

In the sections that follow, we describe the DigitEyes articulated object tracking system in more detail, along with the specific modeling choices required for hand tracking.

3 State Model for Articulated Mechanisms

The state model encodes all possible mechanism configurations and their corresponding image feature patterns as a two-part mapping between state and feature spaces. The first part is a kinematic model which captures all possible spatial link positions, while the second part is a feature model which describes the image appearance of each link shape.

3.1 Kinematic Models: Application to the Human Hand

We model kinematic chains, like the finger, with the Denavit-Hartenburg (DH) representation, which is widely used in robotics [13]. Since feature models require geometric information not captured in the kinematics, the DH description of each link is augmented with an additional transform from the link frame to a *shape frame*, which describes the position of the visible link geometry in space. A solid model in the shape frame generates features through projection into the image.

We model the hand as a collection of 16 rigid bodies: 3 individual finger links (called phalanges) for each of the five digits, and a palm. From a kinematic viewpoint, the hand consists of multi-branched kinematic chains attached to a six DOF base. We make several simplifying assumptions in modeling the hand kinematics. First, we assume that each of the four fingers of the hand are planar mechanisms with four degrees of freedom (DOF). The abduction DOF moves the plane of the finger relative to the palm, while the remaining 3 DOF determine the finger's configuration within the plane. Fig. 1 illustrates the planar finger model. Each finger has an *anchor point*, which is the position of its base joint center in the frame of the palm, which is assumed to be rigid. The base joint is the one farthest (kinematically) from the finger tip. We use a four parameter

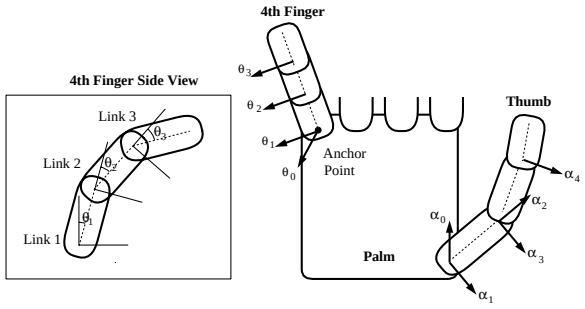

Fig. 1. Kinematic models, illustrated for fourth finger and thumb. The arrows illustrate the joint axes for each link in the chain.

quaternion representation of the palm pose, which eliminates rotational singularities at the cost of a redundant parameter. The total hand pose is described by a 28 dimensional state vector.

The thumb is the most difficult digit to model, due to its great dexterity and intricate kinematics. We currently employ the thumb model used in Rijpkema and Girard's grasp modeling system [12] (see Fig. 1). They were able to obtain realistic animations of human grasps using a five DOF model. DH parameters for the first author's right hand, used in the experiments, can be found in [11].

Real fingers deviate from our modeling assumptions in three ways. First, most fingers deviate slightly from planarity. This deviation could be modeled with additional kinematic transforms, but we have found the planar approximation to be adequate in practice. Second, the last two joints of the finger, counting from the palm outwards, are driven by the same tendon and are not capable of independent actuation. It is simpler to model the DOF explicitly, however, than to model the complicated angular relationship between the two joints. The third and most significant modeling error is change in the anchor points during motion. We have modeled the palm as a rigid body, but in reality it can flex. In gripping a baseball, for example, the palm will conform to its surface, causing the anchor points to deviate from their rest position by tens of millimeters. Fortunately, for free motions of the hand in space, the deviation seems to be small enough to be tolerated by our system.

The modeling framework we employ is general. To track an arbitrary articulated structure, one simply needs its DH parameters and a set of shape models that describe its visual appearance. Within the subproblem of hand tracking, this allows us to develop a suite of hand models whose DOFs are tailored to specific applications.

3.2 Feature Models: Description of Hand Images

The output of the hand state model is a set of features consisting of lines and points generated by the projection of the hand model into the image plane.

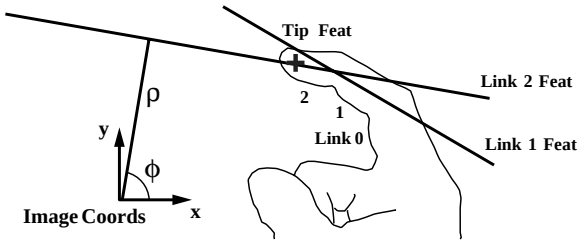

Fig. 2. Features used in hand tracking are illustrated for finger links 1 and 2, and the tip. Each infinite line feature is the projection of the finger link central axis.

Each finger link, modeled by a cylinder, generates a pair of lines in the image corresponding to its occlusion boundaries. The bisector of these lines, which contains the projection of the cylinder central axis, is used as the link feature. The link feature vector $[a\, b\, \rho]$ gives the parameters of the line equation $ax+by-\rho=0$. Using the central axis line as the link feature eliminates the need to model the cylinder radius or the slope of the pair of lines relative to the central axis, which is often significant near the finger tips. We use the entire line because the endpoints are difficult to measure in practice. Fig. 2 shows two link feature lines extracted from the first two links of a finger.

Each finger tip, modeled by a hemisphere, generates a point feature by projection of the center into the image. The finger tip feature vector $[x\, y]$ gives the tip position in image coordinates, as illustrated in Fig. 2. The total hand appearance is described by a $(3m+2n)$-dimensional vector, made up of link and tip features, where m and n are the number of finger links and tips, respectively, in the model.

Other feature choices for hand tracking are possible, but the occlusion contours are the most powerful cue. Hand albedo tends to be uniform, making it difficult to use correlation features. Shading is potentially valuable, but the complicated illuminance and self-shadowing of the hand make it difficult to use.

4 Feature Measurement: Detection of Finger Links and Tips

Local image-based trackers are used to measure hand features. These trackers are the projections of the spatial hand geometry into the image plane, and they serve to localize and simplify feature extraction. A finger link tracker, drawn as a "T"-shape, is depicted along with its measured line feature in Fig. 3. The stem of the "T" is the projection of the cylinder center axis into the image. The image sampling rate ensures that the true feature location is near the projected tracker.

Once the link tracker has been positioned, line features are extracted by sampling the image in slices perpendicular to the central axis. For each slice, the derivative of the 1D image profile is computed. Peaks in the derivative with the

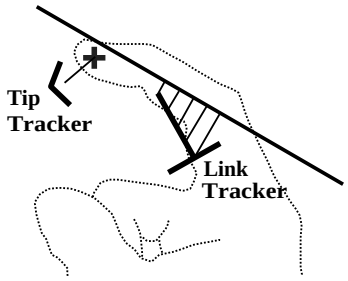

Fig. 3. Image trackers, detected features, and residuals for a link and a tip are shown using the image from Fig. 2. Slashed lines denote the link residual error between the T-shaped tracker and its extracted line measurement. Similarly, the tip tracker (carat shape) is connected to its point feature (cross) by a residual vector.

correct sign correspond to the intersection of the slice with the finger silhouette. The extracted intensity profile and peak locations for a single slice are illustrated in Fig. 4. Line fitting to each set of two or more detected intersections gives a measurement of the projected link axis. If only one silhouette line is detected for a given link, the cylinder radius can be used to extrapolate the axis line location. Currently, the length of the slices (search window) is fixed by hand. Finger tip positions are measured through a similar procedure.

Using local trackers and sampling along lines in the image reduces the pixel processing requirements of feature measurement, permitting fast tracking.

5 State Estimation for Articulated Mechanisms

State estimation proceeds by making incremental state corrections between frames. One cycle of the estimation algorithm goes as follows: The current state estimate is used to predict feature locations in the next frame and position feature trackers. After image acquisition and feature extraction, measured and predicted feature values are compared to produce a state correction, which is added to the current estimate to obtain a new state estimate. The difference between measured and predicted states is modeled by a residual vector, and the state correction is obtained by minimizing its magnitude squared. A high image sampling rate allows us to linearize the nonlinear mapping from state to features around an operating point, which is recomputed at each frame, to obtain a linear least squares problem in the model Jacobian. The following subsections describe the residual model and estimation algorithm in detail.

5.1 Residual Model: Link and Tip Image Alignment

The tip residual measures the Euclidean distance in the image between predicted ($\mathbf{c}_i$) and measured ($\mathbf{t}_i$) tip positions. The residual for the ith tip feature is a

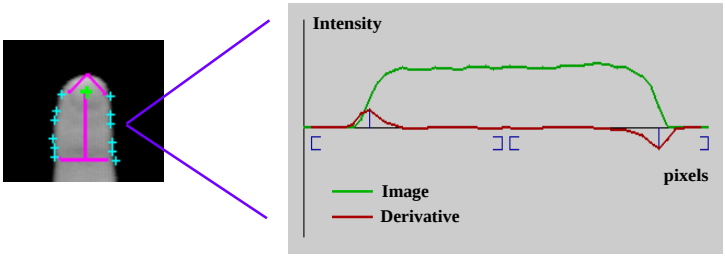

Fig. 4. A single link tracker is shown along with its detected boundary points. One slice through the finger image of a finger is also depicted. Peaks in the derivative give the edge locations.

vector in the image plane defined by

$$\mathbf{v}_i(\mathbf{q}) = \mathbf{c}_i(\mathbf{q}) - \mathbf{t}_i \ , \tag{1}$$

where $\mathbf{c}_i$ is the projection of the tip center into the image as a function of the hand state.

The link residual is a scalar that measures the deviation of the projected cylinder axis from the measured feature line. It is illustrated for a single finger link in Fig. 3. The residual at a point along the axis equals the perpendicular distance to the feature line. We incorporate the orthographic camera model into the residual equation by setting $\mathbf{m} = [a\, b\, 0]^t$ and writing

$$l_i(\mathbf{q}) \ = \ \mathbf{m}^t \mathbf{p}_i(\mathbf{q}) - \rho \ , \tag{2}$$

where $\mathbf{p}_i(\mathbf{q})$ is the 3D position of a point on the cylinder link in camera coordinates, and $[a\ b\ \rho]$ are the line feature parameters. The total link residual consists of one or more point residuals along the cylinder axis (at the base and tip), each given by (2). Note that both residuals are linear in the model point positions.

The feature residuals for each link and tip in the model are concatenated into a single residual vector, $\mathbf{R}(\mathbf{q})$. If the magnitude of the residual vector is zero, the hand model is perfectly aligned with the image data.

5.2 Estimation Algorithm: Nonlinear Least Squares

The state correction is obtained from the residual vector by minimizing $\mathbf{H}(\mathbf{q}) = \frac{1}{2} \| \mathbf{R}(\mathbf{q}) \|^2$. We employ the Levenburg-Marquardt (LM) algorithm for nonlinear least squares problems [3]. The source of nonlinearity in the state model for articulated mechanisms is trigonometric terms in the forward kinematic model. The other source of nonlinearity, inverse depth coefficients in the perspective camera model, is absent in our orthographic formulation.

Let $\mathbf{R}(\mathbf{q}_j)$ be the residual vector for image j. The LM state update equation is given by

$$\mathbf{q}_{j+1} = \mathbf{q}_j - [\mathbf{J}_j^t \mathbf{J}_j + \mathbf{S}]^{-1} \mathbf{J}_j^t \mathbf{R}_j \ , \tag{3}$$

where $\mathbf{J}_j$ is the Jacobian matrix for the residual $\mathbf{R}_j$, both of which are evaluated at $\mathbf{q}_j$. $\mathbf{S}$ is a constant diagonal conditioning matrix used to stabilize the least squares solution. $\mathbf{J}_j$ is formed from the link and tip residual Jacobians. The same basic approach was used by Lowe in his rigid body tracking system [7].

In the remainder of this section, we derive the link Jacobian and discuss its computation. The tip Jacobian derivation proceeds identically, and can be found in [11]. To calculate the link Jacobian we differentiate (2) with respect to the state vector, obtaining

$$\frac{\partial l_i(\mathbf{q})}{\partial \mathbf{q}} \;=\; \mathbf{m}^t \frac{\partial \mathbf{p}_i(\mathbf{q})}{\partial \mathbf{q}} \quad . \tag{4}$$

The above gradient vector for link i is one row of the total Jacobian matrix. Geometrically, it is formed by projecting the *kinematic Jacobian* for points on the link, $\partial \mathbf{p}_i(\mathbf{q})/\partial \mathbf{q}$, in the direction of the feature edge normal.

The kinematic Jacobian in (4) is composed of terms of the form $\partial \mathbf{p}_i/\partial \mathbf{q}_j$, which arise frequently in robot control. As a result, these Jacobian entries can be obtained directly from the model kinematics by means of some standard formulas (see [13], Chapter 5). There are three types of Jacobians, corresponding to joint rotation, spatial translation, and spatial rotation DOFs. All points must be expressed in the frame of the camera producing the measurements. For a revolute (rotational) DOF joint $\mathbf{q}_j$ we have

$$\frac{\partial \mathbf{p}_i}{\partial \mathbf{q}_j} = \mathbf{w}_j \times (\mathbf{p}_i - \mathbf{d}_c^j) \quad , \tag{5}$$

where $\mathbf{w}_j$ is the rotation axis for joint j expressed in the camera frame, and $\mathbf{d}_c^j$ is the position of the joint j frame in camera coords. There will be a similar calculation for *each* camera being used to produce measurements.

The Jacobian calculation for the palm DOFs must reflect the fact that palm motion takes place with respect to the world coordinate frame, but must be expressed in the camera frame. We obtain the rotation and translation components:

$$\frac{\partial \mathbf{p}_i}{\partial \mathbf{v}} = \mathbf{R}_c^w \quad \text{and} \quad \frac{\partial \mathbf{p}_i}{\partial \mathbf{q}_j} = [\mathbf{R}_c^w \mathbf{J}_w]_j \times \mathbf{p}_i \quad , \tag{6}$$

where $\mathbf{v}$ is the palm velocity with respect to the world frame and $\mathbf{q}_j$ is a component of the quaternion specifying palm rotation. In addition, $\mathbf{R}_c^w$ is the camera to world rotation and $\mathbf{J}_w$ is a Jacobian mapping quaternion velocity to angular velocity, with $[\cdot]_j$ denoting the jth column of a matrix.

5.3 Tracking with Multiple Cameras

The tracking framework presented above generalizes easily to more than one camera. When multiple cameras are used, the residual vectors from each camera are concatenated to form a single global residual vector. This formulation can exploit partial observations. If a finger link is visible in one view but not in the another due to occlusion, the single view measurement is still incorporated into the residual, and therefore the estimate.

6 Experimental Results: Hand Tracking With Two Cameras

The DigitEyes system was used to track a full 27 DOF hand model, using two camera image sequences. Because the hand motion must avoid occlusions for successful tracking, the available range of travel is not large. It is sufficient, however, to demonstrate recovery of articulated DOFs in conjunction with palm motion. Figure 5 (at the end of the paper) shows sample images, trackers, and features from both cameras at three points along a 200 frame sequence. The two cameras are set up about a foot and a half apart with optical centers verging near the middle of the tracking area. Fig. 6 shows the estimated model configurations corresponding to the sample points. In the left column, the estimated model is rendered from the viewpoint of the first camera. In the right column, it is shown from an arbitrary viewpoint, demonstrating the 3D nature of our tracking result. The estimated state trajectories for the entire sequence are given in Figs. 7 and 8.

Direct measurement of tracker accuracy is difficult due to the lack of ground truth data. We plan to use a Polhemus sensor to measure the accuracy of the 6 DOF palm state estimate. Obtaining ground truth measurements for joint angles is much more difficult. One possible solution is to wear an invasive sensor, like the DataGlove, to obtain a baseline measurement. By fitting the DataGlove inside a larger unmarked glove, the effect of the external finger sensors on the feature extraction can be minimized.

7 Implementation Details

The DigitEyes system is built around a special board for real-time image processing, called IC40. Each IC40 board contains a 68040 CPU, 5 MB of dual-ported RAM, a digitizer, and a video generator. The key feature of this board is its ability to deliver digitized images to processor memory at video rate with no computational overhead. This removes an important bottleneck in most workstation-based tracking systems. Ordinary C code can be compiled and down-loaded to the board for execution.

In the multicamera implementation, there is an IC40 board for each camera. The total computation is divided into two parts: feature extraction and state estimation. Feature extraction is done in parallel by each board, then the extracted features are passed over the VME bus to a Sun workstation, which combines them and solves the resulting least squares problem to obtain a state estimate. Estimated states are passed over the Ethernet to a Silicon Graphics Indigo 2 workstation for model rendering and display.

8 Conclusion

We have presented a visual tracking framework for high DOF articulated mechanisms, and its implementation in a tracking system called DigitEyes. We have

demonstrated real-time hand tracking of a 27 DOF hand model using two cameras. We will extend this basic work in two ways. First, we will modify our feature extraction process to handle occlusions and complicated backgrounds. Second, we will analyze the observability requirements of articulated object tracking and address the question of camera placement.

References

1. A. Blake, R. Curwen, and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *Int. J. Computer Vision*, 11(2):127–145, 1993.
2. T. Darrell and A. Pentland. Space-time gestures. In *Looking at People Workshop*, Chambery, France, 1993.
3. J. Dennis and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1983.
4. B. Dorner. Hand shape identification and tracking for sign language interpretation. In *Looking at People Workshop, IJCAI*, Chambery, France, 1993.
5. D. Gennery. Visual tracking of known three-dimensional objects. *Int. J. Computer Vision*, 7(3):243–270, 1992.
6. S. B. Kang and K. Ikeuchi. Grasp recognition using the contact web. In *Proc. IEEE/RSJ Int. Conf. on Int. Robots and Sys.*, Raleigh, NC, 1992.
7. D. Lowe. Robust model-based motion tracking through the integration of search and estimation. *Int. J. Computer Vision*, 8(2):113–122, 1992.
8. R. Mann and E. Antonsson. Gait analysis– precise, rapid, automatic, 3-d position and orientation kinematics and dynamics. *BULLETIN of the Hospital for Joint Diseases Orthopaedic Institute*, XLIII(2):137–146, 1983.
9. J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(6):522–536, 1980.
10. A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):730–742, 1991.
11. J. Rehg and T. Kanade. Digiteyes: Vision-based human hand tracking. Technical Report CMU-CS-TR-93-220, Carnegie Mellon Univ. School of Comp. Sci., 1993.
12. H. Rijpkema and M. Girard. Computer animation of knowledge-based human grasping. *Computer Graphics*, 25(4):339–348, 1991.
13. M. Spong. *Robot Dynamics and Control*. John Wiley and Sons, 1989.
14. M. Yamamoto and K. Koshikawa. Human motion analysis based on a robot arm model. In *IEEE Conf. Comput. Vis. and Pattern Rec.*, pages 664–665, 1991. Also see Electrotechnical Laboratory Report 90-46.
15. T. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. A hand gesture interface device. In *Proc. Human Factors in Comp. Sys. and Graphics Interface (CHI+GI'87)*, pages 189–192, Toronto, Canada, 1987.

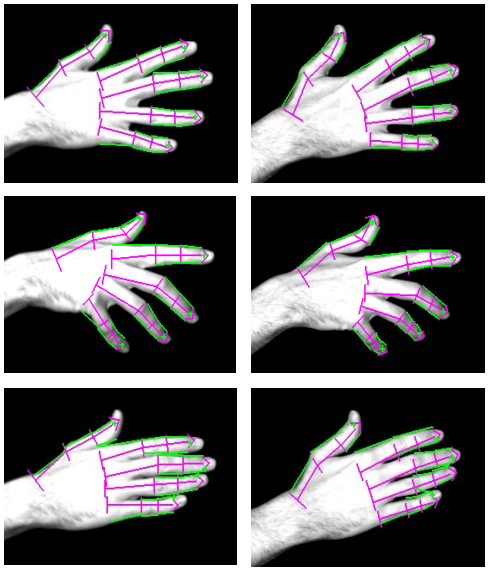

Fig. 5. Three pairs of hand images from the continuous motion estimate plotted in Figs. 7 and 8. Each stereo pair was obtained automatically during tracking by storing every fiftieth image set to disk. The samples correspond to frames 49, 99, and 149.

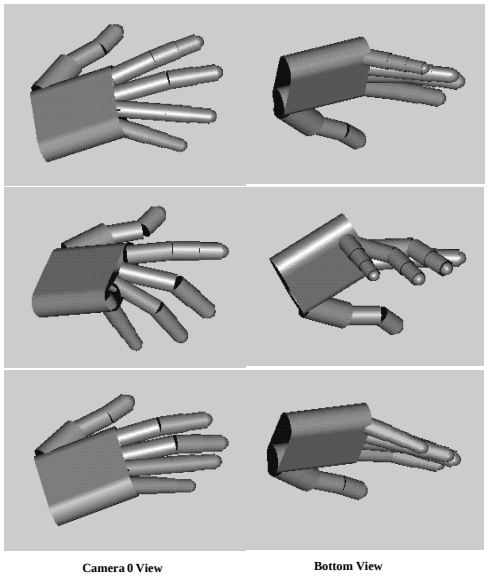

Fig. 6. Estimated hand state for the image samples in Fig. 5, rendered from the Camera 0 viewpoint (left) and a viewpoint underneath the hand (right).

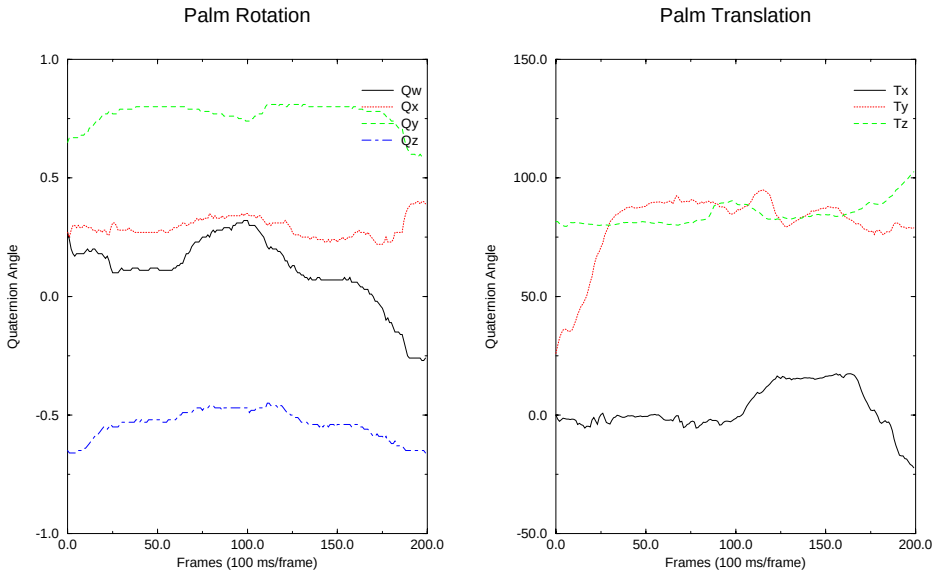

Fig. 7. Estimated palm rotation and translation for motion sequence of entire hand. $\mathbf{Q}_w$-$\mathbf{Q}_z$ are the quaternion components of rotation, while $\mathbf{T}_x$-$\mathbf{T}_z$ are the translation. The sequence lasted 20 seconds.

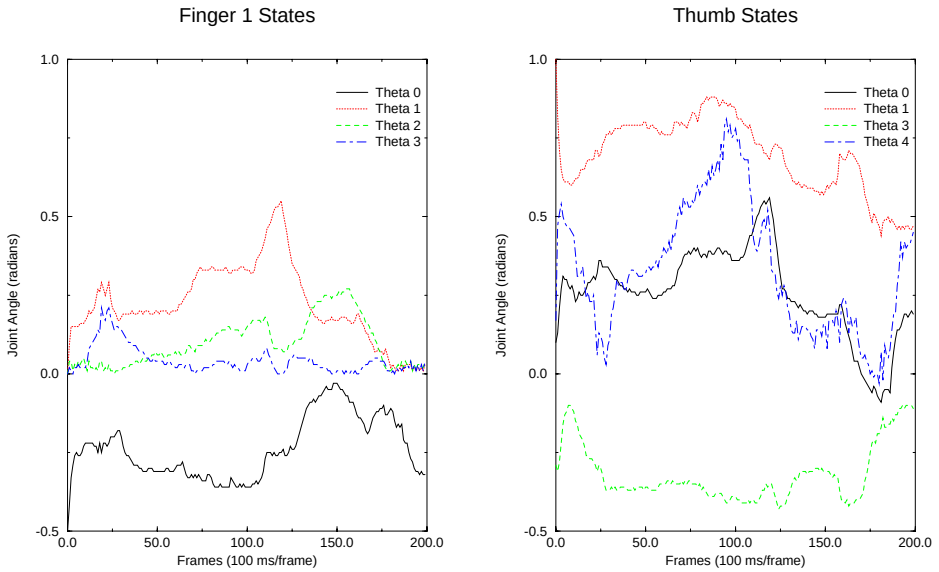

Fig. 8. Estimated joint angles for the first finger and thumb. The other three fingers are similar to the first. Refer to Fig. 1 for variable definitions.