Toward 3-D Gesture Recognition[*]

James Davis[†]
MIT Media Lab
Massachusetts Institute of Technology
Cambridge, MA 02139

Mubarak Shah
Computer Vision Lab
University of Central Florida
Orlando, FL 32826

Abstract

This paper presents a glove-free method for tracking hand movements using a set of 3-D models. In this approach, the hand is represented by five cylindrical models which are fit to the third phalangeal segments of the fingers. Six 3-D motion parameters for each model are calculated that correspond to the movement of the fingertips in the image plane. Trajectories of the moving models are then established to show the 3-D nature of the hand motion.

Keywords Motion Estimation, Hand Tracking, Gesture Recognition.

[*]The research reported here was supported by the National Science Foundation grants CDA-9200369, IRI 9122006, and IRI-9220768.

[†]The author was a member of the Computer Vision Lab at the University of Central Florida during this research.

1 Introduction

The importance of human gestures has been greatly underestimated. We each use hundreds of expressive movements every day [2, 9], with many of these movements pertaining to hand gestures. These movements may have radically different interpretations from country to country – one hand gesture may represent a meaning of "good" in one country, whereas in another country it may be viewed as offensive [9]. Finger-spelling, a subset of sign language, permits any letter of the English alphabet to be presented using a distinct hand gesture. Using the finger-spelling gesture set, people can communicate words to one another using only hand movements [4]. The media has realized the significance of gestures and was experienced in the final scene of the movie, *Close Encounters of the Third Kind* (**Columbia Pictures**, 1977), where a human and alien communicated to each another using hand movements. **McDonald's** demonstrated the utilization of gestures in a 1994 television commercial showcasing patrons ordering any one of four different meals using the appropriate hand gesture. If we are to enhance and extend the man-machine interface, it is imperative to enable computers to interpret hand motions and to act intelligently according to their meanings.

Tracking hand motion becomes more realistic with a 3-D, rather than a 2-D, approach. With 3-D information, we know the real-world location of the fingers at any time, and can exploit this knowledge to suit applications without having to concern ourselves with the weaker and possibly ambiguous 2-D information. Two-dimensional ambiguities which may arise are the 3-D trajectories which, after undergoing perspective projection, have the same corresponding 2-D trajectory. Also, using 3-D models and motion parameters avoids the need for motion correspondence, which attempts to map feature points to their correct 2-D trajectory , for each feature point is a member of a distinct model for a particular finger and thus has no ambiguity in which trajectory it belongs. Therefore to remove these uncertainties which may arise in 2-D, we can look to 3-D.

In this paper, we discuss our method for developing a computer vision system which has the ability to model and track rigid 3-D finger movement of a glove-free hand. Advantages over our previous method include the removal of the glove for fingertip detection, the elimination of motion correspondence, and the use of more meaningful 3-D hand information.

The rest of this paper discusses our approach, which first identifies the fingers of the hand (Section 3.1) and fits a 3-D generalized cylinder to the third phalangeal segment of each finger (Section 3.2). Then six 3-D motion parameters are calculated for each model corresponding to the 2-D movement of the fingers in the image plane (Section 4). Experiments are shown with 3-D hand movements (Section 5). The 3-D motion trajectories of the models are given, which may used in the tracking and recognition of gestures.

2 Related Work

Regh and Kanade [10] describe a model-based hand tracking system called *DigitEyes*. This system uses stereo cameras and special real-time image processing hardware to recover the state of a hand model with 27 spatial degrees of freedom. In order for *DigitEyes* to be used in specific hand applications, the kinematics, geometry, and initial configuration of the hand must be known in advance. Hand features are measured using local image-based trackers within manually selected search windows. Rendered models and state trajectories are given demonstrating the 3-D nature of their results.

Darrell and Pentland [5] have proposed an approach for gesture recognition using sets of 2-D view models of a hand (one or more example views of a hand). These models are matched to stored gesture patterns using dynamic time-warping, where each gesture is warped to make it of the same length as the longest model. Matching is based upon the normalized correlation between the image and the set of 2-D view models. This method requires the use of special-purpose hardware to achieve real-time performance, and uses gray-level correlation which can be highly sensitive to noise.

Cipolla, Okamoto, and Kuno [3] present a structure from motion (SFM) method in which the 3-D visual interpretation of hand movements is used in a man-machine interface. A glove with colored markers is used as input to the vision system and movement of the hand results in motion between the markers in the images. The authors use the affine transformation of an arbitrary triangle formed by the markers to determine the projection of the axis of rotation, change in scale, and cyclotorsion. This information is used to alter the position and orientation of an object displayed on a computer graphics system. The information extracted from the markers does not give the position of each finger, it only provides a triangular reference plane for the SFM algorithm.

Fukumoto, Mase, and Suenaga [6] present a system called *Finger-Pointer* which recognizes pointing actions and simple hand forms. The system uses stereo image sequences to determine the 3-D location of the pointing finger. Their system first locates the coordinates of the operator's fingertip and its pointing direction. A cursor is then displayed in the target position on an opposing screen. The system is robust in that it is able to detect the pointing regardless of the operator's pointing style.

Segan's [11] *Gest* is a computer vision system that learns to identify non-rigid 2-D hand shapes and computes their pose. This system consists of three phases: *data collection*, *learning*, and *recognition*. In *data collection*, the system displays a hand in a fixed position on the screen and the user responds by presenting that same gesture to the camera. *Learning* is executed off-line and attempts to calculate the hand's pose and classify the user's hand gesture. *Recognition* involves graph matching and employs a preclassifier to offset the matching cost. Each gesture is determined from the hand's 2-D position, and does not use any motion characteristics or 3-D feature locations. *Gest* was used to control graphics applications, such as a graphics editor and flight simulator.

Kang and Ikeuchi [8] describe a framework for determining 3-D hand grasps. An intensity image is used for the identification and localization of the fingers using curvature analysis, and a range image is used for 3-D cylindrical fitting of the fingers. Lines were physically drawn on the fingers to help identify particular segments. A *contact web*, a structure comprised of contact points of the hand with the grasped object, is used to map a low-level hand configuration to a more abstract grasp description. The grasp is then identified using a *grasp cohesive index*. The three identifiable phases (*pregrasp*, *grasp*, and *manipulation*) are used to determine a grasping task. The *pregrasp* phase performs the intended grasp without the target object. Here the hand preshape and transportation are calculated. In the *grasp* phase, the hand touches and has a stable hold of the object. The *manipulation* phase contains hand motions and object movement. Though this method uses 3-D finger information, it requires both intensity and costly range imagery to produce the finger models.

3 Finger Modelling

To generate an appropriate 3-D model for the hand, we require only one intensity image of the user's hand in a predefined start position. To begin, we first identify the fingers within the image and determine each finger's axis of orientation. Then generalized cylinders are fit to specific finger segments. Anatomical knowledge of the human hand is exploited to enhance the modelling process.

3.1 Identification of Finger Regions

Initially, we constrain the user to begin with the hand in a known start position (See Fig. 1.a). Using histogram thresholding, the original image is converted into a binary image in which small regions are removed (See Fig.1.b). We then find a set of points which can be used to differentiate the fingers from the rest of the image. Previous approaches for finding feature points involve boundary curvature extrema [8], interest operators to detect specially colored regions [3], and manual selection [13]. Our approach relies on knowledge of the start position and natural design of the hand to automatically determine five fingertip points $\{T_n\}_{n=0}^{4}$ and seven base points $\{B_m\}_{m=0}^{6}$ which are used to segment the fingers. Each finger region is found by applying a connected component algorithm using the respective fingertip and base points as bounds in the segmentation (See Fig. 1.c). We know a priori, due to the required start position and anatomy of the hand, that the middle finger's fingertip (T_2) has the highest y-coordinate of all the fingertips, and that the thumb's fingertip (T_0) has the largest x-coordinate. Given a fingertip $T_n \mid n \geq 2$, if there is a finger to the left in the image, then this left fingertip must be at a lower y-coordinate and smaller x-coordinate position by nature of human hand design (extreme cases as in hand deformities are not considered). Similarly, given a fingertip $T_n \mid n \leq 2$, if there is a finger to the right in the image, then this right fingertip must be at a lower y-coordinate and greater x-coordinate. By first finding T_2 and T_0, we can apply this fingertip knowledge to reduce the search space and easily find the remaining fingertips T_1, T_3, and T_4. To find a base point, we move the fingertip points that lie on either side of the targeted base point down along the inner boundary of the fingers until they converge into the same point. This valley location is the base point. Base points B_1 (using T_0 and T_1), B_3 (using T_1 and T_2), B_4 (using T_2 and T_3), and B_5 (using T_3 and

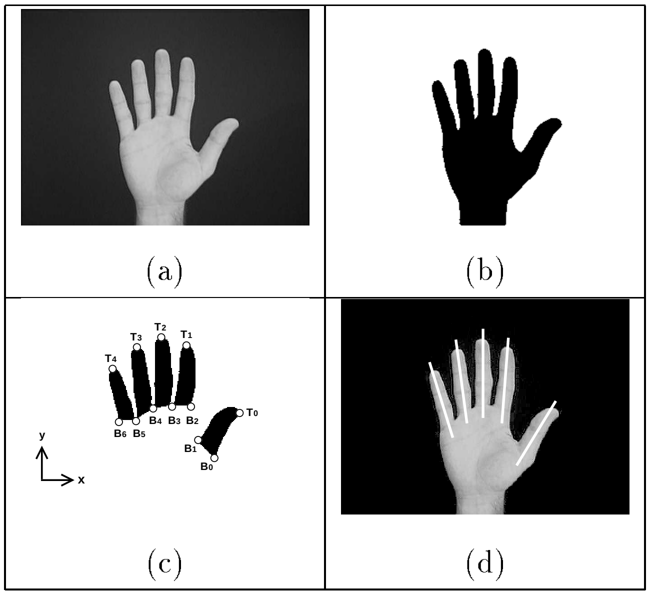

Figure 1: Determining Finger Orientation. (a) Start position of the hand in frame 000. (b) Binary image resulting from histogram thresholding and removal of small regions. (c) Finger regions found using fingertip points $\{T_n\}_{n=0}^{4}$ and base points $\{B_m\}_{m=0}^{6}$. (d) Frame 000 showing each finger's orientation axis.

T_4) are found in this manner. To find base points B_2 and B_6, we level off the base of the respective finger with the x-axis and use the resulting corner as the base point. As for B_0, it can be approximated by moving $-45°$ from B_1 to the opposing side of the thumb. Once the fingers have been identified using these points, the axis of orientation for each finger can be calculated (See Fig. 1.d). The orientation axis is established by finding the line in which the integral of the square of the distance to points in the finger is a minimum. The integral to be minimized over finger F is

$$E = \int\int_F r^2 \, dx \, dy \quad , \tag{1}$$

where r is the perpendicular distance from point (x, y) to the axis sought after [7]. The fingers and axes will be used in generating cylindrical representations of finger segments.

3.2 Cylindrical Fitting

Cylindrical models can be employed to represent the fingers due to the inherent cylindrical nature of fingers. A finger as a whole is a non-rigid object, with the first phalangeal (FP), second phalangeal (SP), and third phalangeal (TP) segments (only FP and TP segments for thumb) [12] each exhibiting rigid behavior. If the fingers were to be modeled in their entirety, three independent phalangeal models for each finger would be required due to the non-rigidness of fingers. Also, if all three segments were to be modelled, special concerns arise to ensure the spatial connectedness of the three phalangeal models while deriving the independent motions of the segments. Occlusion then becomes a major problem, for the FP and SP segments can be occluded much of the time. Restricting the user to rigid finger movement would allow one generalized cylinder to be fit to the entire finger. If this were the case, only one section, e.g. TP segment, need be modelled to reduce the computational overhead, and then this target area, e.g. fingertip, can be tracked throughout the sequence. Using only the TP segments also reduces the spatial relation and occlusion problem. (Issues concerning non-rigid behavior and motion misinterpretation due to particular motions are discussed in Section 4.3.) Therefore, for simplicity, models representing only the TP segments are used to track the movements of each fingertip. To model the TP segments, we must know where they are located with respect to each finger in the image. In general, each FP, SP, and TP segment length occupies nearly a third of the total finger length. Using this heuristic, the major axis for the finger can be divided into three parts (except for the thumb, where it is divided into two), designating the TP segment as the upper most third of the finger (upper half for the thumb) along the axis of orientation. A straight homogeneous generalized cylinder (SHGC)[13, 14] can then be fit to give a 3-D model to each 2-D TP segment (See Fig. 2.a&b), such that each model's projection conforms to the actual respective fingertip in the image (See Fig. 2.c). Since the angle between the cross-section plane and the SHGC axis (orientation axis or spine) is $90°$, a more precise definition of *right* SHGCs (RSHGCs) is used [14]. A cross-section shape of an ellipse is used to fit the natural cross-section of a finger, with semi-major axis a and semi-minor axis b, having $b = f(a) \mid f(a) < a$. When fitting the ellipse cross-sections near the fingertip, semi-major axis a becomes increasingly smaller. Since $b = f(a)$, the two ellipse axes will be in proportion to one another resulting in closure of the cylinder into a realistic 3-D fingertip-like appearance (See Fig. 2.a). When generating the cylinder for the thumb, it must be rotated to correspond to the real 3-D orientation of the thumb, such that semi-major axis a makes a $45°$ angle with the XY plane through the hand.

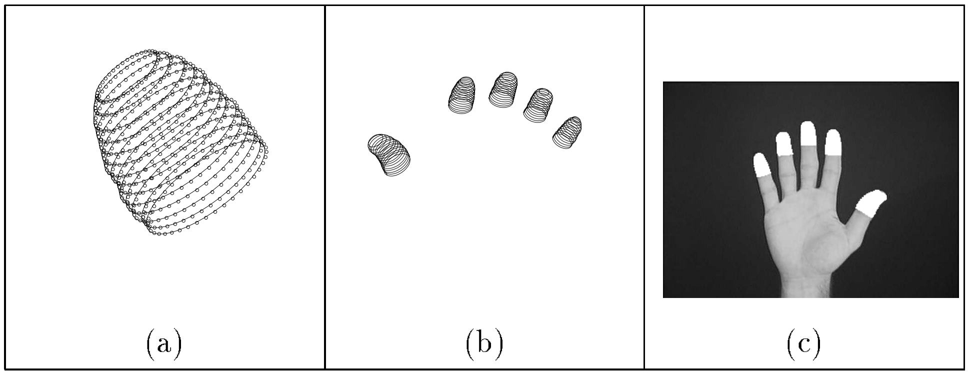

Figure 2: TP Models. (a) Index finger's 3-D cylindrical TP model shown with nodes. (b) All five TP models representing a model set for the hand. (c) Projection of models (in white) onto the hand in the image.

4 Motion Parameter Estimation

Given a set of TP models and a sequence of intensity images in which the hand is moving, we would like compute the 3-D motion of the fingertips employing the 2-D motion in the image plane. The 3-D motion of a model is represented in terms of translation (T_x, T_y, T_z) and counter-clockwise rotation $(\omega_x, \omega_y, \omega_z)$ around the three coordinate axes based at the model's centroid. Our approach incorporates a direct method using spatio-temporal derivatives instead of optical flow, a linearized rotation matrix (due to small motion changes), and a 3-D model (where the depth is known) to compute the 3-D motion. An over constrained set of equations is established and solved for the unknown motion parameters. With slight enhancements to the algorithm to cope with multiple frame estimation, the locations of the TP models can be continually updated in 3-D location to match the 2-D fingertip movement.

4.1 Choosing Visible Model Nodes

A 3-D TP model is comprised of visible nodes (facing the viewing plane) and occluded nodes (located on the model's back-side and facing away from the viewing plane). Nodes which are occluded cannot be used in the motion parameter calculation, for they do not correspond to any point in the image plane.

We can determine the visibility of nodes by using together two methods for back-side elimination [1]. To begin, the 3-D surface normal **n** for each node is compared with the

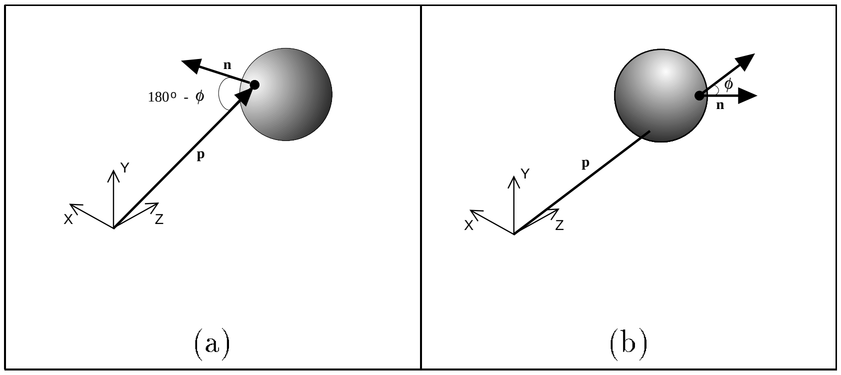

Figure 3: Determining Point (Node) Visibility. (a) Angle ϕ between point vector **p** and surface normal **n** is $> 90°$ (or equivalently, $180° - \phi$ is $< 90°$); the point is facing the viewing plane. (b) Angle ϕ is $< 90°$; the point faces away from the viewing plane and is occluded.

node's point vector **p**. If the angle ϕ between the two vectors is $\geq 90°$, then the surface normal is pointing toward the viewing plane and the node is labeled as *possibly visible* (See Fig. 3.a). If angle ϕ is $< 90°$, then the surface normal is pointing away from the viewing plane and the node is occluded and discarded (See Fig. 3.b). This alone is not enough to determine the visibility of nodes for if the model contains a large number of nodes, many *possibly visible* nodes may project onto the same pixel in the image plane. To reduce this redundancy and have a one-to-one mapping of nodes to pixels, the *possibly visible* nodes are projected and stored in a depth array where each cell in the array corresponds to a unique pixel in the image plane. If two or more nodes project onto the same cell, the node with the smallest depth (closest) is retained, and the other node(s) are discarded. After all the *possibly visible* nodes are projected, only those nodes that remain in the depth array are labeled as *visible* and are used in the motion estimation process. Calculating the visible model nodes using surface normals and a depth array gives an accurate representation of the model which can be seen in the image plane. This process must be performed each time the model location is updated to ensure that previously visible nodes have not become occluded and vice-versa.

4.2 Formulation of Motion Parameter Estimation

Consider the optical flow constraint equation:

$$f_x u + f_y v + f_t = 0 \ , \tag{2}$$

where $f_x = \frac{\partial f}{\partial x}$, $f_y = \frac{\partial f}{\partial y}$, $f_t = \frac{\partial f}{\partial t}$, $u = \frac{dx}{dt}$, and $v = \frac{dy}{dt}$. Assume that the geometry projection from 3-D space onto the 2-D image plane is perspective projection with camera focal length F. Then the optical flow field (u, v) induced by the 3-D instantaneous motion about the object centroid is given by:

$$u = \frac{F}{Z}\left[(T_x + \omega_y Z_c - \omega_z Y_c) + \frac{-X}{Z}(T_z + \omega_x Y_c - \omega_y X_c)\right] \quad , \tag{3}$$

$$v = \frac{F}{Z}\left[(T_y + \omega_z X_c - \omega_x Z_c) + \frac{-Y}{Z}(T_z + \omega_x Y_c - \omega_y X_c)\right] \quad , \tag{4}$$

where (T_x, T_y, T_z) is the forward translation vector, $(\omega_x, \omega_y, \omega_z)$ is the counter-clockwise rotation vector, (X, Y, Z) are the world coordinates, and (X_c, Y_c, Z_c) are the object centered coordinates.

Substituting the above equations for u and v in (2) and rearranging, we get

$$\begin{aligned} -f_t \;=\;& f_x \frac{F}{Z}\left[(T_x + \omega_y Z_c - \omega_z Y_c) + \frac{-X}{Z}(T_z + \omega_x Y_c - \omega_y X_c)\right] \\ &+ f_y \frac{F}{Z}\left[(T_y + \omega_z X_c - \omega_x Z_c) + \frac{-Y}{Z}(T_z + \omega_x Y_c - \omega_y X_c)\right] \quad , \end{aligned} \tag{5}$$

which can also be written as

$$\begin{aligned} -f_t \;=\;& \left[f_x \frac{F}{Z}\right] T_x + \left[f_y \frac{F}{Z}\right] T_y - \left[\frac{F}{Z^2}(f_x X + f_y Y)\right] T_z \\ & - \left[\frac{F}{Z^2}(f_x X Y_c + f_y Z Z_c + f_y Y Y_c)\right] \omega_x \\ & + \left[\frac{F}{Z^2}(f_x Z Z_c + f_x X X_c + f_y Y X_c)\right] \omega_y \\ & - \left[\frac{F}{Z}(f_x Y_c - f_y X_c)\right] \omega_z \quad . \end{aligned} \tag{6}$$

In this equation, (X, Y, Z) and (X_c, Y_c, Z_c) are known from the model, and f_x, f_y, and f_t can be computed from image pairs. Therefore the only unknowns are the motion parameters (T_x, T_y, T_z) and $(\omega_x, \omega_y, \omega_z)$. An over constrained set of equations is established using visible nodes and in matrix form is as follows

$$[\mathbf{A}]\,\mathbf{x} = \mathbf{b} \quad ,$$

with $\mathbf{x} = (T_x, T_y, T_z, \omega_x, \omega_y, \omega_z)^T$. A linear regression using least squares is used to approximate the six unknown motion parameters in $\mathbf{x}$, and is iterated to account for linearizing.

4.3 Motion Estimation Conditions

For successful tracking with this implementation, the hand motion must be small and avoid occlusions. It is also important to calculate f_x, f_y, and f_t with sub-pixel accuracy to keep the projected nodes from moving randomly within a local neighborhood. Spatial-temporal 3×3 Sobel masks were used to compute f_x and f_y, and locations with small gradients cannot be used for motion estimation and are excluded from the regression to yield a more stable estimate.

After each estimation, the model nodes are updated to their new location. Previously visible nodes which have become occluded are excluded from the next iteration, and previously occluded nodes which become visible may be used if they were utilized in a previous estimation. Convergence can be determined by analyzing the root-mean-square error of the intensity difference $(-f_t)$ vector.

To reduce the error accumulation associated with multiple frame estimations, the visible nodes with intensity and gradient information from the first image are propagated throughout the sequence. Initially, for calculating the motion parameters between frame 1 and frame 2, the visible model nodes record the corresponding intensity and gradient information from frame 1. Then the motion parameters are determined using the model nodes and frame 2. After application of the parameters to the model from frame 1, the model is now located to conform to frame 2. For frame 3, a new estimation is calculated using the model (compensated from frame 1 to frame 2) and frame 3. This process continues, propagating the intensity and gradient information from frame 1 through the remainder of the sequence until either significant accumulated rotation causes the gradients to change, large displacement from the original frame changes the intensity values, or too few original visible nodes remain in the model. If any of these cases occur, the model from the previous estimation is re-projected onto its corresponding frame to gather new information. In general, This procedure segments one long sequence of images into a set of smaller length sequences, each having its own local intensity and gradient propagation.

5 Experiments

Our system was used to track two distinct hand movements: movement in the XY plane (See Fig. 4), movement in the XZ plane, i.e. scaling (See Fig. 5), These examples are sufficient to demonstrate the advantage of a 3-D, rather than a 2-D, approach. In each sequence, the locations of the TP models were updated in each frame to match the movement of the fingertips in the image plane (See superimposed models in Figs 4&5). In sequence 1, with no depth changes, the 2-D trajectories are shown to be sufficient to approximate the motion of the hand (Compare 2-D and 3-D trajectories in Figs. 4). Sequence 2 demonstrates the hand changing in depth. This type of motion can be shown in 3-D (See 3-D trajectories in Fig. 5) and cannot be distinguished in 2-D, where it appears that the hand is mainly at rest (See 2-D trajectories in Fig. 5).

As for gesture recognition, we performed Spock's well known "Live Long and Prosper" hand gesture from **Star Trek** to the system, which tracked the hand from the start position to the fixed gesture position (See Fig. 6). The resulting calculated movements can then be used in gesture recognition methods.

6 Conclusion

In this paper, we presented a 3-D hand modelling and motion estimation method for tracking hand movements. This approach does not require any glove or motion correspondence, and recovers 3-D motion information of the hand. The orientation of the fingers in a 2-D image are found, and a generalized cylinder is fit to each finger's third phalangeal segment. Six motion parameters for each finger are calculated, which correspond to the 2-D movement of the fingertips in the image plane. Three-dimensional trajectories are then determined from the motion of the models, which may be used in hand tracking and gesture recognition applications.

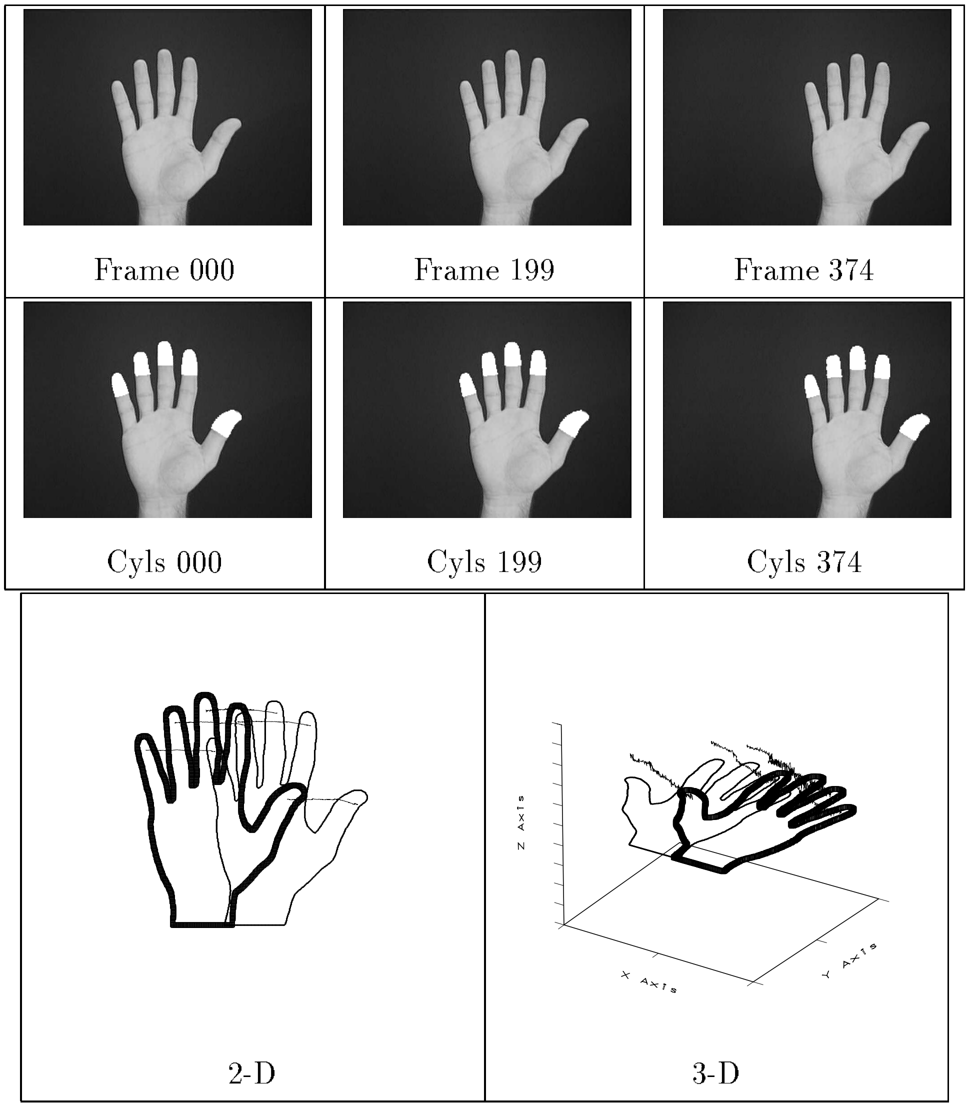

Figure 4: Sequence 1. First Row: Sampled images from a sequence where the hand translates in the XY plane. Second Row: Images from first row superimposed with the projection of TP models (shown in white). Third Row: 2-D and 3-D trajectories (hand outline and models represent the initial and final hand positions, respectively).

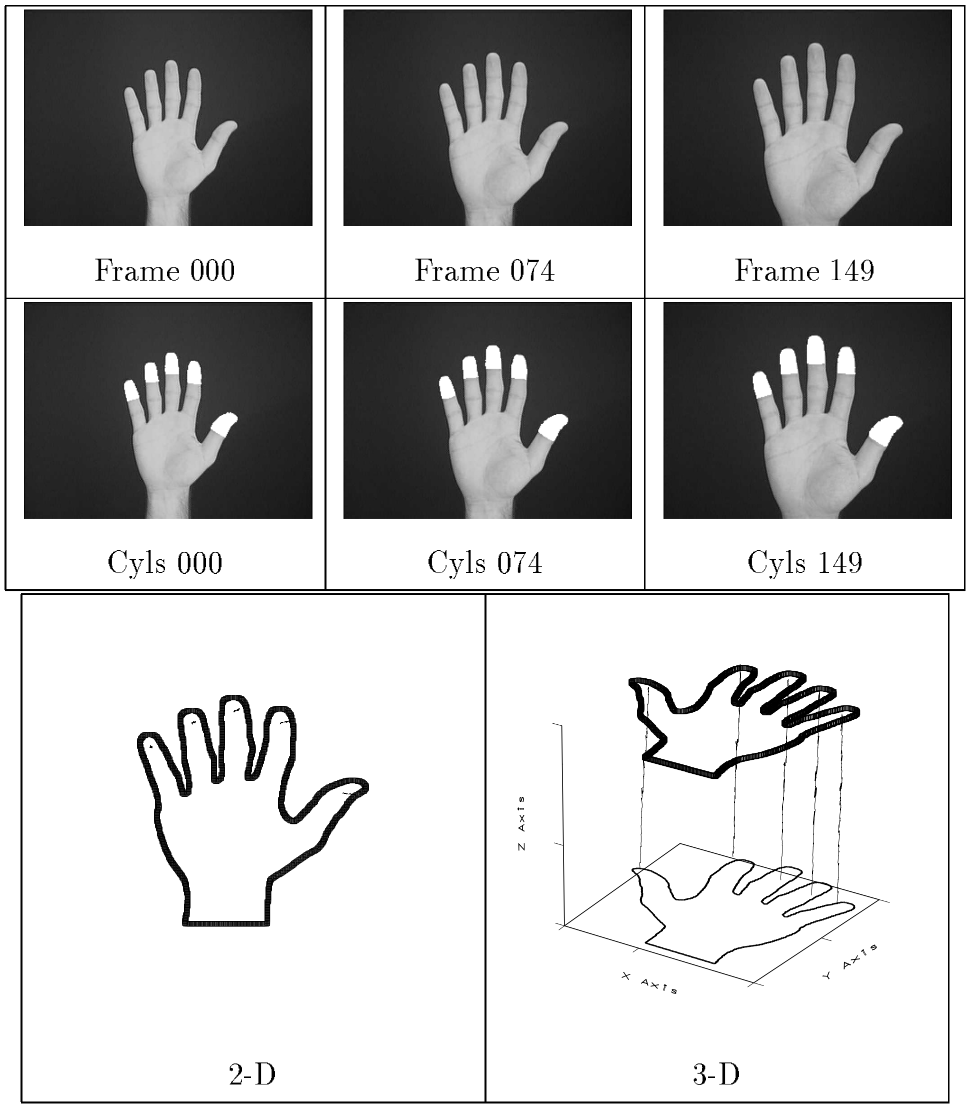

Figure 5: Sequence 2. First Row: Sampled images from a sequence where the hand translates in the XZ plane, i.e. scaling. Second Row: Images from first row superimposed with the projection of TP models (shown in white). Third Row: 2-D and 3-D trajectories (hand outline and models represent the initial and final hand positions, respectively).

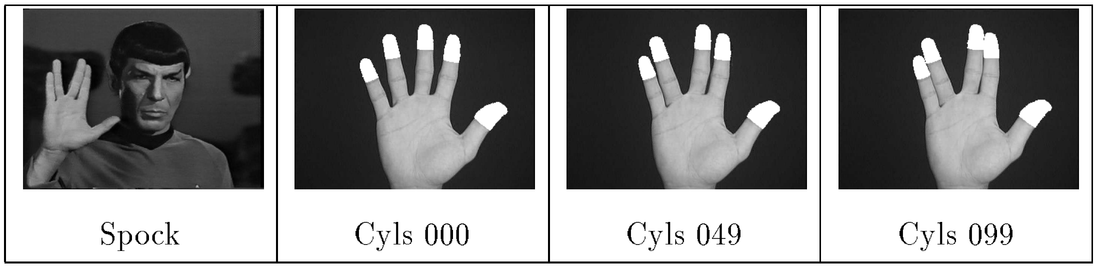

Figure 6: Spock's "Live Long and Prosper" hand gesture. Spock making the classic hand gesture (**Star Trek**: *Amok Time*, 1967), and the corresponding image sequence superimposed with the updated locations of the TP models (shown in white). The sequence begins with the hand in the start position and ends with the hand in the recognizable gesture position.

References

[1] Artwick, B. *Applied Concepts in Microcomputer Graphics.* Prentice-Hall, New Jersey, 1984.

[2] Bauml, B., and Bauml, F. *A Dictionary of Gestures.* The Scarecrow Press, New Jersey, 1975.

[3] Cipolla, R., Okamoto, Y., and Kuno, Y. Robust structure from motion using motion parallax. In *ICCV*, pages 374–382. IEEE, 1993.

[4] E. Costello. *Signing: How to Speak With Your Hands.* Bantam Books, New York, 1983.

[5] Darrell, T., and Pentland, A. Space-time gestures. In *CVPR*, pages 335–340. IEEE, 1993.

[6] Fukumoto, M., Mase, K., and Suenaga, Y. Real-time detection of pointing actions for a glove-free interface. In *IAPR Workshop on Machine Vision Applications*, pages 473–476, December 1992.

[7] Horn, B.K.P. *Robot Vision.* McGraw-Hill, 1986.

[8] Kang, S.B., and Ikeuchi, K. Toward automatic robot instruction from perception – recognizing a grasp from observation. *IEEE Transactions of Robotics and Automation*, 9:432–443, August 1993.

[9] Morris, D., Collet, P., Marsh, P., and O'Saughnessy, M. *Gestures: Their Origins and Distribution.* Stein and Day, 1979.

[10] Rehg, J., and Kanade, T. Visual tracking of high dof articulated structures: an application to human hand tracking. In *ECCV*, pages 35–46, May 1994.

[11] Segen, J. Gest: A learning computer vision system that recognizes hand gestures. *Machine Learning IV*, 1994.

[12] Taylor, C., and Schwarz, R. The anatomy and mechanics of the human hand. *Artificial Limbs*, 1955.

[13] Ulupinar, F., and Nevatia, R. Shape from contour: Straight homogeneous generalized cones. In *ICCV*, 1990.

[14] Zerroug, M., and Nevatia, R. Segmentation and recovery of shgcs from a real intensity image. In *ECCV*, 1994.

James W. Davis received his BS degree with honors in computer science from the University of Central Florida in 1994 and a MS degree from the MIT Media Laboratory in 1996. He is currently a PhD candidate at the MIT Media Laboratory, where his research interests include the modeling and recognition of human and animal movement, motion understanding, gesture recognition, and human-computer interfaces.