Teleoperating ROBONAUT: A case study

Geovanni Martinez[1], Ioannis A. Kakadiaris[1] and Darby Magruder[2]

[1]Visual Computing Lab
Department of Computer Science
University of Houston
Houston TX 77204-3010
Email:{geovanni,ioannisk}@uh.edu

[2]Robotic Systems Technology Branch
Automation and Robotics Division
NASA - Johnson Space Center
Houston, TX-77058
Email: darby.f.magruder1@jsc.nasa.gov

Technical Report UH-CS-02-01

Department of Computer Science

University of Houston

March 2002

This work was supported by an ISSO Postdoctoral Fellowship and a software grant (NDDS) from Real Time Innovations, Inc.

Abstract

In this paper, we present a non-intrusive method for human motion estimation from a monocular video camera for the teleoperation of ROBONAUT (ROBOtic astroNAUT). ROBONAUT is an anthropomorphic robot developed at NASA - JSC, which is capable of dextrous, human- like maneuvers to handle common extravehicular activity tools. The human operator is represented using an articulated three-dimensional model consisting of rigid links connected by spherical joints. The shape of a link is described by a triangular mesh and its motion by six parameters: one three-dimensional translation vector and three rotation angles. The motion parameters of the links are estimated by maximizing the conditional probability of the frame-to-frame intensity differences at observation points. The algorithm was applied to synthetic and real test sequences of a moving arm with very encouraging results. Specifically, the mean error for the derived wrist position (using the estimated motion parameters) was 0.6 ± 1.0 mm for the synthetic image sequences and 0.57 ± 0.31 cm for the real test sequences. The motion estimates were used to remotely command a robonaut simulation developed at NASA - JSC.

1 Introduction

While the microgravity environment of space provides numerous opportunities it also poses challenges that must be overcome. In particular, space systems operations and maintenance for the Human Exploration and Development of Space will demand a heavy extravehicular activity (EVA) workload from a small number of crew members. For example, close to 900 EVA hours will be required to assemble the International Space Station with an additional 200 hours per year for maintenance requirements. Hence, robotic devices remotely working with supervised or teleoperated control will be needed to alleviate the astronaut work load as much as possible. It is expected that by the end of the next decade the majority of the EVA required operations on orbit and on planetary missions will be conducted with the assistance of telerobotic devices. Telerobotic technology seeks to merge robotics and teleoperation to develop robots with remote mobility and manipulation. In such a system, the human operator is physically removed from the task, sends commands to the robot over a telecommunication system, and receives information about the status of the task and its environment using sensors.

Hence, telerobotics requires a strong interaction between the human operator and the controlled robot with the aid of vision, tactile and force feedback sensors [21].

One such system is the ROBONAUT (ROBOtic astroNAUT) which is an anthropomorphic robot with two arms, two hands, a head, a torso and a stabilizing leg, that is currently being developed at NASA - Johnson Space Center (NASA - JSC) to provide an astronaut substitute for EVA operations (Fig. 1). It includes two 7 degree-of-freedom (DOF) arms, two 12 DOF five-finger robot hands, a 6+ DOF leg and a head with a 4 DOF stereo camera. The robot arms are capable of dextrous, human-like maneuvers to handle common EVA tools. The ROBONAUT will be teleoperated by an intravehicular crew using sensors, cameras and virtual reality tools (head mounted display, virtual reality gloves, or force-reflective arm and hand masters). One intuitive way to teleoperate the ROBONAUT is to estimate the three-dimensional motion of the operator's body parts (e.g., head, arms, torso, and legs) and then use the estimated motion to control the ROBONAUT. In such a system, the robot duplicates the movements made by an operator. For example, as the operator extends out an arm, so does the ROBONAUT. And if the operator starts twisting a screwdriver, the ROBONAUT should duplicate the action. Currently, the off-the-shelf systems for human motion estimation are very intrusive and encumbering because they attach devices such as sensors or markers to the operator. Our goal is to develop a non-intrusive system for human motion estimation from a monocular image sequence for the teleoperation of ROBONAUT [21].

The existing literature on human motion estimation from a monocular image sequence can be roughly divided into two groups (see [1, 11, 25, 26, 36] for comprehensive reviews). The first one estimates the motion using image features (e.g, edge points) [8, 10, 12, 13, 16, 17, 18, 20, 24, 35, 37, 38, 40, 41, 42, 43, 44]. The second group estimates the motion from frame to frame intensity differences at observation points [7, 19, 27, 28, 29, 39, 45, 47]. Those motion parameters, which minimize the frame to frame intensity differences at observation points, are considered to be the estimates of the motion parameters. In [9, 46] both image features and frame to frame intensity differences are taken into account for motion estimation. In this report, the motion is estimated by maximizing the conditional probability of the frame to frame intensity difference at observation points [32, 33, 34].

For Maximum-Likelihood motion estimation the human body is represented by a three-dimensional model consisting of rigid links connected by

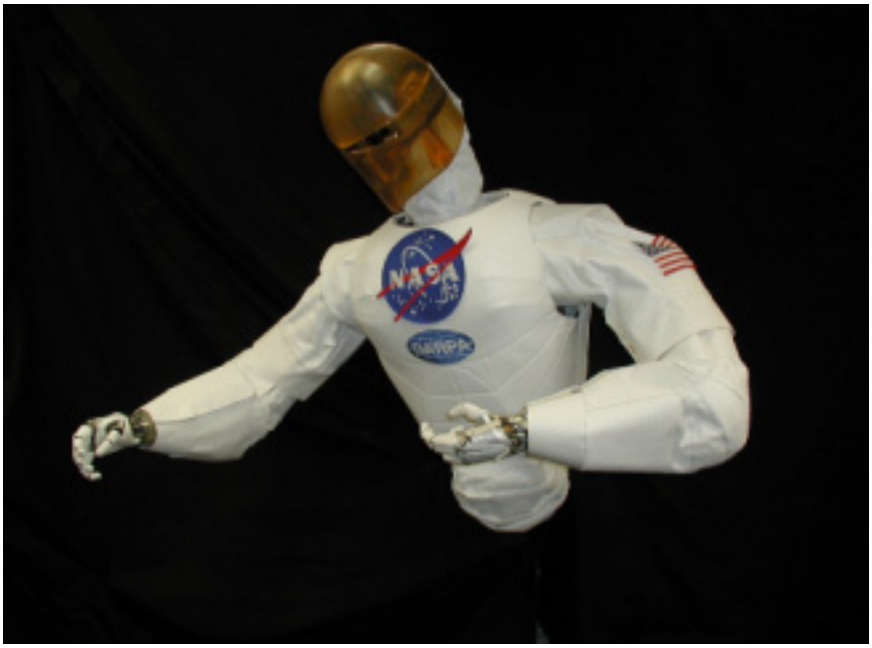


Figure 1: ROBONAUT (Photo courtesy of NASA - JSC)

spherical joints. The three-dimensional (3D) shape of a link is described by a triangular mesh. The 3D motion of a link is described by six parameters: one three-dimensional translation vector and three rotation angles. The texture of a link is defined by projecting a real image into its triangular mesh. The motion parameters of a link are estimated by maximizing the conditional probability of the frame to frame intensity differences at observation points. The conditional probability is a function of the motion parameters, the frame-to-frame intensity differences, and the covariance matrix of the intensity error at the observation points. The intensity error is the result of the camera noise, the shape estimation error, and the position error due to the motion estimation errors occurred by the motion analysis of previous frames. The covariance matrix of the intensity error is computed by modeling the position and the shape estimation error of the link and the camera noise by zero-mean stationary Gaussian stochastic processes. In addition, instead of simultaneously estimating all the motion parameters of the links, a decomposition approach is used. Thus, first the translation and rotation parameters of the root link are estimated. Then only the rotation angles for the rest of the links are estimated beginning from the root link one after the other.

In order to improve the accuracy and reliability of the motion estimates, for each link the Maximum-Likelihood estimator is applied iteratively.

Until now, the Maximum-Likelihood motion estimation has been only applied to estimate the motion parameters of the head and shoulders of a subject. In this work, we develop a model of the right arm of a human and apply the Maximum-Likelihood motion estimation. Then, we employ the motion estimates to remotely command the right arm of a virtual ROBONAUT using a simulation developed at NASA - Johnson Space Center [15]. Finally, we perform a number of experiments on synthetic and real data to assess the accuracy, limitations and advantages of the approach.

The remainder of this paper is structured as follows. In Section 2, the Maximum-Likelihood motion estimation algorithm is described. In Section 3, the process for commanding the right arm of the virtual ROBONAUT is presented. In Section 4, experimental results for synthetic and real image sequences are detailed. Finally, in Section 5, we offer our conclusions.

2 Maximum-Likelihood Motion Estimation

In this section, we will describe the Maximum-Likelihood motion estimation algorithm of articulated objects proposed in [32, 34], that we use for estimating the motion of the right arm of a human body from a monocular image sequence. For each link of a human body, the algorithm estimates the parameters, which describe its 3D motion in the world coordinate system from time t_k to time t_{k+1}. For estimation of the motion parameters of a link, the frame to frame intensity differences of two consecutive intensity frames I_k and I_{k+1} of an image sequence, are evaluated. Those motion parameters, which maximize the conditional probability of the frame to frame intensity differences at observation points, are considered to be the motion estimates of the link. The conditional probability is a function of the motion parameters, the frame to frame intensity differences, and the covariance matrix of the intensity error at the observation points. To compute this conditional probability, a mathematical relationship between the 3D motion parameters of the link and the frame to frame intensity differences at the observation points is used. This relationship is based on a number of assumptions about the world and how it is projected into the image plane of a camera as described in [32] and reviewed in section 2.1. Section 2.2 defines what is an observation point and explains how to compute the corresponding frame to frame intensity dif-

ference. Sections 2.3 and 2.4 describe how to compute the covariance matrix of the intensity errors and the conditional probability of the frame to frame intensity differences at observation points, respectively. Section 2.5 explains how to maximize the conditional probability taking the derivative respect to the motion parameters. Finally, section 2.6 presents a stepwise description of the Maximum-Likelihood motion estimation algorithm.

2.1 World model

The world model summarizes the assumptions about how the world is constructed and how it is projected into the image plane of a camera. It allows us to establish the connection between the frame to frame intensity differences and the moving links [32]. It consists of an illumination model, an object model, and a camera model. The illumination model assumes that the illumination is diffuse as well as spatial and time invariant.

2.1.1 Object model

The object model is composed of a shape model, a material model, and a motion model.

The *shape model* describes the human body as M rigid links L_m, $m = 0, .., M-1$, connected to each other by $M-1$ spherical joints J_j, $j = 1, .., M-1$. A spherical joint J_j is represented by a point between the two connected links and its position is described by the position vector $\mathbf{J}_j$. The shape of each link is described by a triangular mesh (Fig. 2). Each link has its local coordinate system and the position of the vertices of its triangular mesh are expressed w.r.t. this coordinate system. The two links connected by an arbitrary joint J_j are named reference link $L_{c(j)^+}$ and relative link $L_{c(j)^-}$ of the joint J_j. The reference link $L_{c(j)^+}$ (one of the two links connected by the joint J_j) is the link that is closer to the root link L_0 in the tree structure. The functions $c(j)^+$ and $c(j)^-$ give the identification number m of the corresponding reference link and relative link of the joint J_j, respectively. For example, the value of these functions for the spherical joint J_1 in Fig. 2 is $c(1)^+ = 0$ and $c(1)^- = 1$. In addition, the origin of the coordinate system of the relative link $L_{c(j)^-}$ is placed at the joint position $\mathbf{J}_j$ and the origin of the coordinate system of the root link is placed at an arbitrary point $\mathbf{J}_0$ inside its mesh. Finally, the links along a branch of the tree structure are numbered increasingly beginning from the root link one after the other and

each joint gets the number of the corresponding relative link. Due to this regular numbering the following convenient identities are true: $c(j)^- = j$, and $L_{c(j)^-} = L_j$.

The *material model* assumes that the objects have a diffuse reflecting surface and that their texture is the result of a linear combination of the intensity and chrominance values being reflecting from the object surface.

The *motion model* assumes that from time t_k to time t_{k+1} the root link L_0 of an articulated object can rotate and translate freely in the world coordinate system and that the relative link L_j of each joint J_j can only rotate freely around the joint position $\mathbf{J}_j$. The motion of the root link L_0 from time t_k to time t_{k+1} is described first by a rotation and then by a translation of its local coordinate system in the world coordinate system (Fig. 3). The translation is described by the 3D translation vector $\Delta\mathbf{T}_0 = (\Delta T_x^0, \Delta T_y^0, \Delta T_z^0)$. The rotation is described by the rotation matrix $\Delta\mathbf{R}_0$ defined by the three rotation angles $\Delta\omega_x^0$, $\Delta\omega_y^0$, and $\Delta\omega_z^0$. Let $\mathbf{A}$ and $\mathbf{A}'$ the position of an arbitrary point on the surface of the root link L_0 before and after the motion (i.e., at times t_k and t_{k+1}, respectively). The new position $\mathbf{A}'$ is computed as follows:

$$\mathbf{A}' = \Delta\mathbf{R}_0 \cdot (\mathbf{A} - \mathbf{J}_0) + \mathbf{J}_0 + \Delta\mathbf{T}_0 \ , \tag{1}$$

where $\mathbf{J}_0$ is the origin of the coordinate system of the root link L_0 at time t_k. In contrast, the motion of an arbitrary link L_j from discrete time t_k to t_{k+1} is described first by a translation $\Delta\mathbf{T}_j$ and then by a rotation $\Delta\mathbf{R}_j$ of its local coordinate system w.r.t. the world coordinate system (Fig. 3). Due to the constraints imposed by the joint J_j on the motion of the corresponding relative link L_j, the translation vector $\Delta\mathbf{T}_j$ depends entirely on the rotation $\Delta\mathbf{R}_{c(j)^+}$ and translation vector $\Delta\mathbf{T}_{c(j)^+}$ of the corresponding reference link $L_{c(j)^+}$ and is computed as follows:

$$\Delta\mathbf{T}_j = \Delta\mathbf{R}_{c(j)^+} \cdot (\mathbf{J}_j - \mathbf{J}_{c(j)^+}) + \mathbf{J}_{c(j)^+} + \Delta\mathbf{T}_{c(j)^+} - \mathbf{J}_j \ . \tag{2}$$

According to Eq. 2 the motion of the reference link $L_{c(j)^+}$ enforces a translation on the corresponding relative link L_j. Let $\mathbf{A}$ and $\mathbf{A}^\gamma$ be the position of an arbitrary point on the surface of a relative link L_j before the motion (i.e., at time t_k), and after the translation enforced by the corresponding reference link $L_{c(j)^+}$, respectively. The position $\mathbf{A}^\gamma$ is computed as follows:

$$\mathbf{A}^\gamma = \mathbf{A} + \Delta\mathbf{T}_j \ . \tag{3}$$

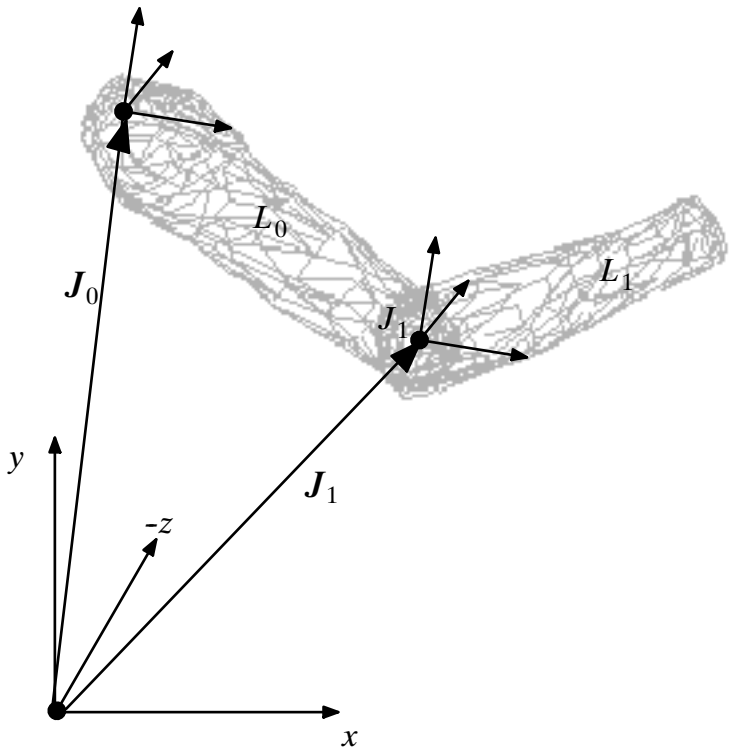


Figure 2: Graphical model of a human arm with two links L_0 and L_1 connected by one spherical joint J_1. The shape of each link is described by a triangular mesh. The links L_0 and L_1 represent the reference and the relative link of the joint J_1, respectively. In this example, the reference link is also the root link of the articulated object. The origin of the coordinate system of the relative link L_1 is placed at the joint position $\mathbf{J}_1$ and the origin of the coordinate system of the root link L_0 is placed at an arbitrary position $\mathbf{J}_0$ inside its mesh.

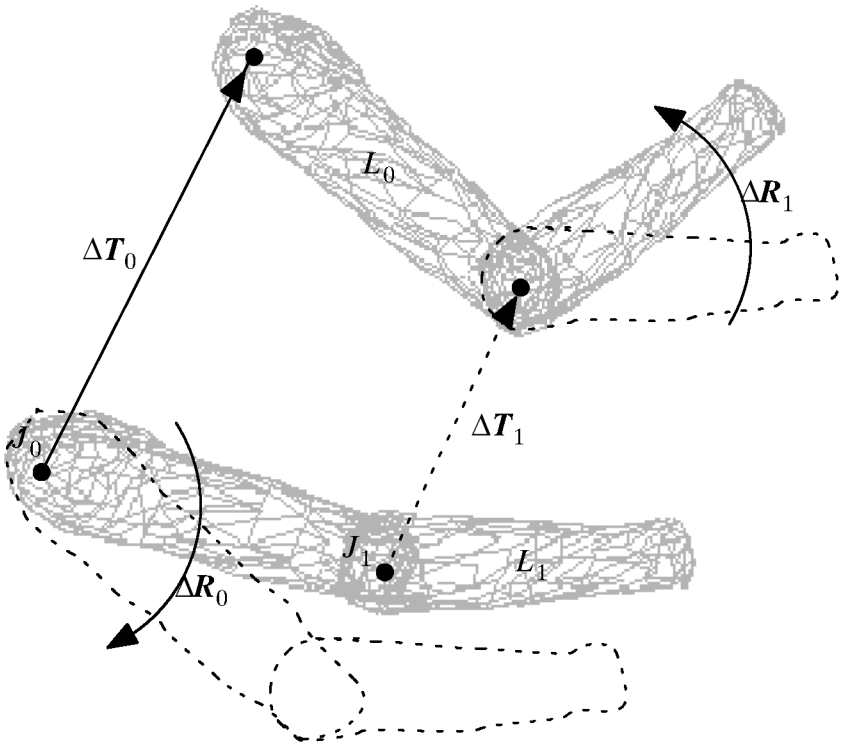


Figure 3: Example of the motion of the human arm model. The motion of the root link L_0 is described first by a rotation $\Delta\mathbf{R}_0$ and then by a translation $\Delta\mathbf{T}_0$ of its local coordinate system in the world coordinate system. The motion of the relative link L_1 is described first by a translation $\Delta\mathbf{T}_1$ and then by a rotation $\Delta\mathbf{R}_1$ of its local coordinate system in the world coordinate system. Due to the constraints imposed by the joint J_1 on the motion of the relative link L_1 its translation vector $\Delta\mathbf{T}_1$ depends entirely on the motion of the corresponding reference link L_0 according to Eq. 2.

The joint position also moves from $\mathbf{J}_j$ to $\mathbf{J}_j^\gamma$ according to the latter equation. Let $\mathbf{A}'$ the position of an arbitrary point on the surface of a relative link L_j after the motion, i.e., at time t_{k+1}. $\mathbf{A}'$ is computed from $\Delta\mathbf{R}_j$ as follows:

$$\mathbf{A}^{'} = \Delta\mathbf{R}_j \cdot (\mathbf{A}^\gamma - \mathbf{J}_j^\gamma) + \mathbf{J}_j^\gamma \ . \tag{4}$$

According to the latter equation the following identity is true: $\mathbf{J}_j{}^{'} = \mathbf{J}_j{}^\gamma$.

2.1.2 Camera model

The camera model assumes that an image is generated by perspective projection of the world into the image plane of a camera (Fig. 4). Thus, an arbitrary point with world coordinates $\mathbf{A} = (A_x, A_y, A_y)^\top$ is projected into the image coordinates $(a_x(\mathbf{A}), a_y(\mathbf{A}))^\top = (f \cdot \frac{A_x}{A_z}, f \cdot \frac{A_y}{A_z})^\top$, where f is the focal length of the camera. The image plane is assumed to be located on the plane described by $z = -f$ (parallel to the xy-plane of the world coordinate system) and the focal point is placed at the origin of the world coordinate system.

The Taylor series expansion of $a_x(\mathbf{A})$ and $a_x(\mathbf{A})$ around an arbitrary point $\mathbf{A}$ are given by:

$$a_x(\mathbf{A} + \Delta\mathbf{A}) = a_x(\mathbf{A}) + \left.\frac{\partial a_x}{\partial A_x}\right|_{\mathbf{A}} \Delta A_x + \left.\frac{\partial a_x}{\partial A_y}\right|_{\mathbf{A}} \Delta A_y + \left.\frac{\partial a_x}{\partial A_z}\right|_{\mathbf{A}} \Delta A_z + \theta_1 \ ,$$

$$a_y(\mathbf{A} + \Delta\mathbf{A}) = a_y(\mathbf{A}) + \left.\frac{\partial a_y}{\partial A_x}\right|_{\mathbf{A}} \Delta A_x + \left.\frac{\partial a_y}{\partial A_y}\right|_{\mathbf{A}} \Delta A_y + \left.\frac{\partial a_y}{\partial A_z}\right|_{\mathbf{A}} \Delta A_z + \theta_2 \ ,$$

where θ_1 and θ_2 represents higher order terms. Considering the linear terms only, the following relationship between a small displacement $\Delta\mathbf{A}$ and its corresponding projection $\Delta\mathbf{a}$ into the image plane is given:

$$a_x(\mathbf{A} + \Delta\mathbf{A}) = a_x(\mathbf{A}) + \begin{bmatrix} \frac{f}{A_z} & 0 & \frac{-f \cdot A_x}{A_z^2} \\ 0 & \frac{f}{A_z} & \frac{-f \cdot A_y}{A_z^2} \end{bmatrix} \cdot \Delta\mathbf{A} \ ,$$

$$\Delta\mathbf{a} = \mathbf{K} \cdot \Delta\mathbf{A} \ . \tag{5}$$

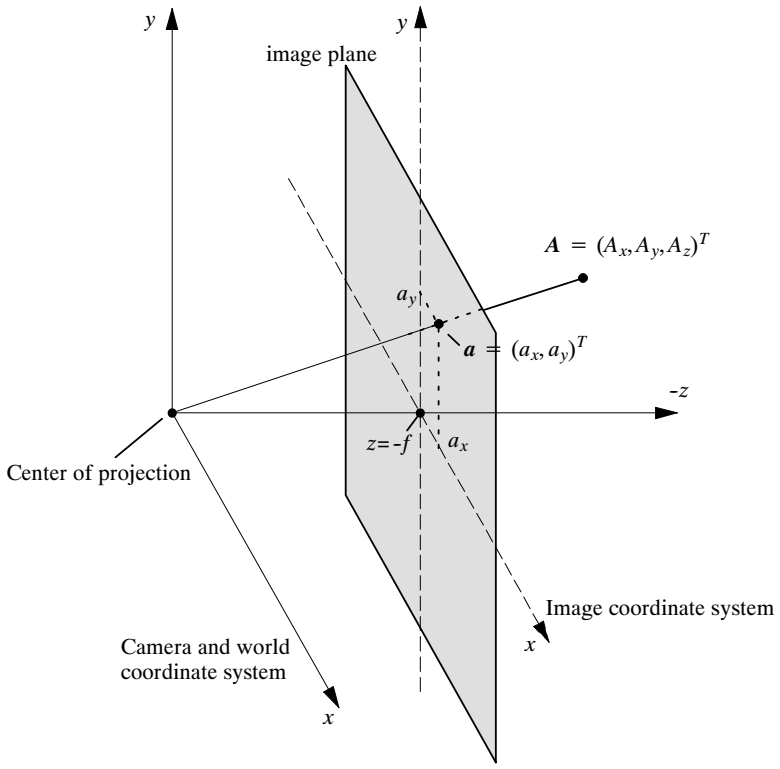


Figure 4: Image coordinates **a** of the perspective projection of an arbitrary point **A** onto the image plane of a camera.

2.2 Observation points

To estimate the motion parameters of an arbitrary link the frame to frame intensity differences at observation points are evaluated. According to [32, 34] an observation point A lies on the surface of the mesh of the link at position $\mathbf{A} = (A_x, A_y, A_y)^\top$ and carries the intensity value I at this position. The position of the projection of this point A into the image plane is represented by $\mathbf{a} = (a_x, a_y)^\top$. Let $\mathbf{g} = (g_x, g_y)^\top$ be the observable linear intensity gradient at image position $\mathbf{a}$. In order to reduce the influence of the camera noise and to increase the accuracy of the estimates only those observation points with high linear intensity gradient ($|\mathbf{g}| > \delta_1$) are taken into account for motion estimation. Assuming that the shape, position and orientation of the model link correspond with those of the real link at time t_k, the frame to frame intensity difference fd at the observation point $\mathbf{a}$ is approximated as follows:

$$fd(\mathbf{a}) = I_{k+1}(\mathbf{a}) - I_k(\mathbf{a}) \approx I_{k+1}(\mathbf{a}) - I \ ,$$

where $I_k(\mathbf{a})$ and $I_{k+1}(\mathbf{a})$ represent the intensity value of the images I_k and I_{k+1} at the position $\mathbf{a}$, respectively. Since in general $\mathbf{a}$ lies outside of the image raster, the intensity value $I_{k+1}(\mathbf{a})$ is computed by bilinear interpolation of the intensity values of the nearest four pixels of the intensity image I_{k+1}. Then, the frame to frame intensity difference at N observation points is represented as follows:

$$\mathbf{FD} = (fd(\mathbf{a}^{(N-1)}), fd(\mathbf{a}^{(N-2)}), \ldots, fd(\mathbf{a}^{(0)}))^\top \ .$$

Finally, the mean squared frame to frame intensity difference at the observation points is given by:

$$msd \ = \ \frac{1}{N} \sum_{n=0}^{N-1} fd(\mathbf{a}^{(n)})^2 \ .$$

2.3 Covariance matrix of the intensity error

In [32, 34] the intensity error at an observation point is assumed to be the result of the camera noise, the shape estimation error, and the position error due to the motion estimation errors occurred by the motion analysis of previous frames. The intensity error at an observation point affects the motion estimation of an arbitrary link because it perturbs the frame to frame intensity differences which are evaluated for motion estimation. The result of

this perturbation is a decrease of the accuracy and reliability of the motion estimates. In this section, we present a stochastic model of the intensity error and a method to compute the covariance matrix of the intensity error at the observation points [32, 34]. In section 2.5, this covariance matrix is taken into account for motion estimation to allow reliable Maximum-Likelihood estimates despite the intensity errors.

The stochastic model for the intensity error is obtained by mapping a stochastic model of the shape estimation and the position errors (due to previous motion estimation errors) into an intensity error on the image plane. Therefore, both the shape and position errors are described by the position error $\Delta\mathbf{A}$, which is the difference between the position of the observation point $\mathbf{A}^*$ on the surface of a real link and the position $\mathbf{A}$ of the corresponding observation point on the surface of the mesh of the link (Fig. 5):

$$\Delta\mathbf{A} = \mathbf{A}^* - \mathbf{A} \ .$$

$\Delta\mathbf{A} = (\Delta A_x, \Delta A_y, \Delta A_y)$ is modeled as stationary zero-mean Gaussian stochastic process with variances σ_x^2, σ_y^2 and σ_z^2. Their covariance matrix is given by:

$$E[\Delta\mathbf{A} \cdot \Delta\mathbf{A}^\top] = C_{\Delta\mathbf{A}} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \ .$$

The position error $\Delta\mathbf{A}$ is mapped into the corresponding position error $\Delta\mathbf{a} = [\Delta a_x, \Delta a_y]^\top = \mathbf{a}^* - \mathbf{a}$ on the image plane using Eq. 5. Δa_x and Δa_y are modeled as stationary zero-mean Gaussian stochastic processes with covariance matrix:

$$\mathbf{C}_{\Delta\mathbf{a}} = E[\Delta\mathbf{a} \cdot \Delta\mathbf{a}^\top] = \mathbf{K} \cdot \mathbf{C}_{\Delta\mathbf{A}} \cdot \mathbf{K}^\top \ ,$$

where

$$\mathbf{C}_{\Delta\mathbf{A}} = \begin{bmatrix} \left(\frac{\sigma_x^2 f^2}{A_z^2} + \frac{\sigma_z^2 f^2 A_x^2}{A_z^4}\right) & \frac{\sigma_z^2 f^2 A_x A_y}{A_z^4} \\ \frac{\sigma_z^2 f^2 A_x A_y}{A_z^4} & \left(\frac{\sigma_y^2 f^2}{A_z^2} + \frac{\sigma_z^2 f^2 A_y^2}{A_z^4}\right) \end{bmatrix} \ .$$

Finally, the position error $\Delta\mathbf{a}$ is mapped into the corresponding intensity error on the image plane $\Delta I_{\Delta\mathbf{a}} = I(\mathbf{a}^*) - I(\mathbf{a})$. Therefore, the intensity signal is approximated using a Taylor series around the position $\mathbf{a}$ on the image plane as follows: $\Delta I_{\Delta\mathbf{a}} = \mathbf{g}^\top \cdot \Delta\mathbf{a}$, where the terms of higher order

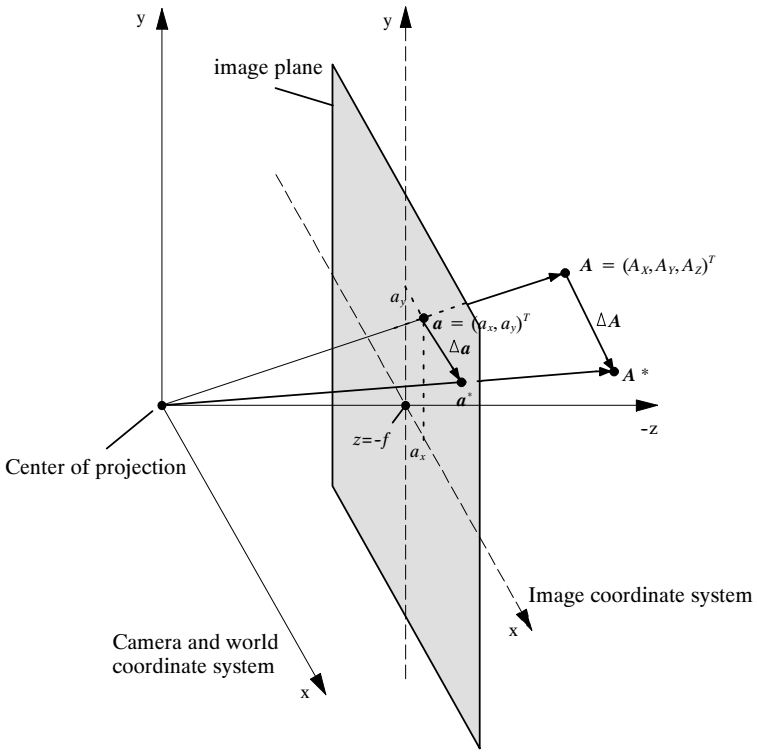


Figure 5: Mapping of a position error $\Delta\mathbf{A}$ in the world coordinate system into a position error $\Delta\mathbf{a}$ on the image coordinate system. $\mathbf{A}^*$ is the position of an observation point on the surface of the link and $\mathbf{A}$ is the corresponding position on the surface of the link's model.

were neglected. This equation describes the changes of the intensity signal I_k from the position $\mathbf{a}$ to the position $\mathbf{a}^*$ on the image plane. The intensity error $\Delta I_{\Delta\mathbf{a}}$ is modeled by a stationary zero-mean Gaussian stochastic process and its variance is computed as follows:

$$\begin{aligned}\sigma^2_{\Delta I_{\Delta\mathbf{a}}} = E[\Delta I_{\Delta\mathbf{a}} \cdot \Delta I^\top_{\Delta\mathbf{a}}] &= \mathbf{g}\cdot\mathbf{C}_{\Delta\mathbf{a}}\cdot\mathbf{g}^\top = \mathbf{g}\cdot\mathbf{K}\cdot\mathbf{C}_{\Delta\mathbf{A}}\cdot\mathbf{K}^\top\cdot\mathbf{g}^\top \\ &= \frac{\sigma_z^2}{A_z^2}\cdot\left((g_x a_x + g_y a_y)^2 + \frac{f^2}{\sigma_z^2}\cdot\left(g_x^2\sigma_x^2 + g_y^2\sigma_y^2\right)\right) .\end{aligned}$$

In addition to $\Delta\mathbf{A}$, the camera noise also generates an intensity error ΔI_{noise} at position $\mathbf{a}$ on the image plane. This intensity error is modeled by a stationary zero-mean Gaussian stochastic process with variance $\sigma^2_{\Delta I_{noise}}$. $\Delta I_{\Delta\mathbf{a}}$ and ΔI_{noise} are assumed to be statistically independent. The total intensity error ΔI at image position $\mathbf{a}$ is assumed to be the sum of both intensity errors with the following variance: $\sigma^2_{\Delta I} = \sigma^2_{\Delta I_{\Delta\mathbf{a}}} + \sigma^2_{\Delta I_{noise}}$.

The joint probability density of the intensity error at N observation points with image coordinates $\mathbf{a}^{(n)}$, $n = 0, .., N-1$, is computed as follows:

$$p_{\mathbf{V}}(\mathbf{V}) = \frac{1}{\sqrt{(2\pi)^N\,|\mathbf{U}|}}\cdot e^{-\frac{1}{2}(\mathbf{V}^\top\cdot\mathbf{U}^{-1}\cdot\mathbf{V})}, \tag{6}$$

where $\mathbf{V} = \left[\Delta I^{(n-1)}, \Delta I^{(n-2)}, \ldots, \Delta I^{(0)}\right]^\top$ is the vector with the N intensity errors, and $|\mathbf{U}|$ is the determinant of the covariance matrix $\mathbf{U}$ of the intensity error at the N observation points. Considering the intensity errors as statistically independent this covariance matrix is expressed as follows:

$$\mathbf{U} = E[\mathbf{V}\cdot\mathbf{V}^\top] = \begin{bmatrix} \sigma^2_{\Delta I^{(n-1)}} & 0 & 0 & \cdots & 0 \\ 0 & \sigma^2_{\Delta I^{(n-2)}} & 0 & \ldots & 0 \\ 0 & 0 & \sigma^2_{\Delta I^{(n-3)}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \sigma^2_{\Delta I^{(0)}} \end{bmatrix} . \tag{7}$$

2.4 Conditional probability of the intensity differences

The motion parameters $\mathbf{B} = (\Delta T_x, \Delta T_y, \Delta T_z, \Delta\omega_x, \Delta\omega_y, \Delta\omega_z)^\top$ of an arbitrary link are estimated by maximizing the conditional probability $p(\mathbf{FD}|\mathbf{B})$ of the frame to frame intensity differences $\mathbf{FD}$ at N observation points. In this section, we reviewed how the conditional probability $p(\mathbf{FD}|\mathbf{B})$ can

be computed from a linear model of the intensity signal at the observation points, the model world and the covariance matrix of the intensity errors at the observation points [32, 34].

Let's consider an arbitrary observation point A on the link surface and assume that from the time t_k to the time t_{k+1} this observation point moves from $\mathbf{A}$ to $\mathbf{A}'$. The corresponding projections into the image plane are $\mathbf{a}$ and $\mathbf{a}'$, respectively. Expanding the intensity signal I_k at image position $\mathbf{a}$ by a Taylor series and neglecting the nonlinear terms, the following relationship between the unknown position $\mathbf{A}'$ and the frame to frame intensity difference is obtained:

$$fd(\mathbf{a}) = I_{k+1}(\mathbf{a}) - I_k(\mathbf{a}) = -\mathbf{g} \cdot (\mathbf{a}^{'} - \mathbf{a}) \ .$$

Replacing $\mathbf{a}$ and $\mathbf{a}'$ with their corresponding coordinates at the world coordinate system the following equation results:

$$fd(\mathbf{a}) = -f \cdot \mathbf{g} \cdot \left(\frac{A_x^{'}}{A_z^{'}} - \frac{A_x}{A_z}, \frac{A_y^{'}}{A_z^{'}} - \frac{A_y}{A_z} \right)^{\top} \ , \tag{8}$$

where the known position $\mathbf{A} = (A_x, A_y, A_z)^{\top}$ is related with the unknown position $\mathbf{A}' = (A_x', A_y', A_z')^{\top}$ according to Eq. 1 as follows: $\mathbf{A}' = \Delta\mathbf{R} \cdot (\mathbf{A} - \mathbf{J}) + \mathbf{J} + \Delta\mathbf{T}$. Substituting $\mathbf{A}'$ in Eq. 8, a highly nonlinear equation that relates the unknown motion parameters $\mathbf{B}$ and the frame to frame intensity difference $fd(\mathbf{a})$ is obtained. This nonlinear equation is linearized in three steps. First, the rotation angles $\Delta\omega_x$, $\Delta\omega_y$, $\Delta\omega_z$ are assumed to be small and thus $\cos\omega \approx 1$ and $\sin\omega \approx \omega$. Second, Eq. 8 is expanded using a Taylor series expansion. Finally, by neglecting the nonlinear terms, the following linear equation that relates the unknown motion parameters and the frame to frame intensity difference is obtained:

$$fd(\mathbf{a}) = -\frac{f \cdot g_x}{A_z} \cdot \Delta T_x - \frac{f \cdot g_y}{A_z} \cdot \Delta T_y + \frac{f \cdot (A_x g_x + A_y g_y)}{A_z^2} \cdot \Delta T_z + Q \ , \tag{9}$$

$$\begin{aligned} Q = & \frac{f \cdot [A_x g_x (A_y - J_y) + A_y g_y (A_y - J_y) + A_z g_y (A_z - J_z)]}{A_z^2} \cdot \Delta\omega_x - \\ & \frac{f \cdot [A_y g_y (A_x - J_x) + A_x g_x (A_x - J_x) + A_z g_x (A_z - J_z)]}{A_z^2} \cdot \Delta\omega_y + \\ & \frac{f \cdot [g_x (A_y - J_y) + g_y (A_x - J_x)]}{A_z} \cdot \Delta\omega_z \ . \end{aligned} \tag{10}$$

When only the rotation parameters of the link need to be estimated, then the equation $fd(\mathbf{a}) = Q$ is used instead of Eq. 9. Both equations can be written as follows:

$$fd(\mathbf{a}) = \mathbf{o}^\top \cdot \mathbf{B} + \Delta I \ , \tag{11}$$

where the term ΔI represents the intensity error caused by the camera noise, the shape estimation error, and the position error due to the motion estimation errors occurred by the motion analysis of previous frames, and

$$\mathbf{o} = \begin{bmatrix} -\frac{f \cdot g_x}{\mathbf{A}_z} \\ -\frac{f \cdot g_y}{\mathbf{A}_z} \\ \frac{f \cdot (A_x g_x + A_y g_y)}{A_z^2} \\ \frac{f \cdot [A_x g_x (A_y - J_y) + A_y g_y (A_y - J_y) + A_z g_y (A_z - J_z)]}{A_z^2} \\ -\frac{f \cdot [A_y g_y (A_x - J_x) + A_x g_x (A_x - J_x) + A_z g_x (A_z - J_z)]}{A_z^2} \\ \frac{f \cdot [g_x (A_y - J_y) + g_y (A_x - J_x)]}{A_z} \end{bmatrix}^\top . \tag{12}$$

When the equation $fd(\mathbf{a}) = Q$ is used, $\mathbf{o}$ is written as follows:

$$\mathbf{o} = \begin{bmatrix} \frac{f \cdot [A_x g_x (A_y - J_y) + A_y g_y (A_y - J_y) + A_z g_y (A_z - J_z)]}{A_z^2} \\ -\frac{f \cdot [A_y g_y (A_x - J_x) + A_x g_x (A_x - J_x) + A_z g_x (A_z - J_z)]}{A_z^2} \\ \frac{f \cdot [g_x (A_y - J_y) + g_y (A_x - J_x)]}{A_z} \end{bmatrix}^\top . \tag{13}$$

Evaluating Eq. 11 at N observation points the following system of linear equations is obtained:

$$\mathbf{FD} = \mathbf{O} \cdot \mathbf{B} + \mathbf{V} \ . \tag{14}$$

Substituting $\mathbf{V} = \mathbf{FD} - \mathbf{O} \cdot \mathbf{B}$ in Eq. 6 the conditional probability of the frame to frame intensity differences at the N observation points is written as follows:

$$p(\mathbf{FD}|\mathbf{B}) = \frac{1}{\sqrt{(2\pi)^N \, |\mathbf{U}|}} e^{-\frac{1}{2}\left((\mathbf{FD}-\mathbf{O}\cdot\mathbf{B})^\top \mathbf{U}^{-1} (\mathbf{FD}-\mathbf{O}\cdot\mathbf{B})\right)} \ . \tag{15}$$

2.5 Maximizing the conditional probability

The motion parameters $\mathbf{B}$ of an arbitrary link are estimated by maximizing the conditional probability $p(\mathbf{FD}|\mathbf{B})$ of the frame to frame intensity differences $\mathbf{FD}$ at N observation points:

$$p(\mathbf{FD}|\widehat{\mathbf{B}}) \geq p(\mathbf{FD}|\mathbf{B}) \quad \forall \mathbf{B} \ , \tag{16}$$

where $\widehat{\mathbf{B}} = (\widehat{\Delta T_x}, \widehat{\Delta T_y}, \widehat{\Delta T_z}, \widehat{\Delta \omega_x}, \widehat{\Delta \omega_y}, \widehat{\Delta \omega_z})^\top$ are the estimated motion parameters. To simplify the maximization the former equation can be written as follows [32, 34]:

$$\frac{\partial \ln p(\mathbf{FD}|\mathbf{B})}{\partial \mathbf{B}}|_{\mathbf{B}=\widehat{\mathbf{B}}} = \frac{\partial((\mathbf{FD} - \mathbf{O} \cdot \mathbf{B})^\top \mathbf{U}^{-1}(\mathbf{FD} - \mathbf{O} \cdot \mathbf{B}))}{\partial \mathbf{B}}|_{\mathbf{B}=\widehat{\mathbf{B}}} = 0 \ .$$

Thus, the Maximum-Likelihood motion estimates $\widehat{\mathbf{B}}$ are given by:

$$\widehat{\mathbf{B}} = \left(\mathbf{O}^\top \mathbf{U}^{-1} \mathbf{O}\right)^{-1} \mathbf{O}^\top \mathbf{U}^{-1} \mathbf{FD} \ . \tag{17}$$

2.6 Algorithm

In this section, we present a step-wise description of the Maximum-Likelihood motion estimation for the human arm. In summary, in the first step the shape, position and orientation of the model is initialized and points on the surface of the model's links are selected as observation points for motion estimation. Second, from each pair of consecutive intensity frames I_k and I_{k+1} the translation vector $\Delta \mathbf{T}_0 = (\Delta T_x^0, \Delta T_y^0, \Delta T_z^0)^\top$ and the rotation angles $\Delta \omega_x^0$, $\Delta \omega_y^0$, and $\Delta \omega_z^0$ of the root link L_0 are estimated. Third, the rotation angles $\Delta \omega_x^j$, $\Delta \omega_y^j$, and $\Delta \omega_z^j$ of the rest of the links L_j, $j = 1, ..., M-1$, are estimated beginning from the root link L_0 one after the other. First, we describe how model is adapted at the beginning of the image sequence and how the observation points are selected for motion estimation. Then, we present the algorithm for Maximum-Likelihood motion estimation of the root link. Finally, we describe the algorithm for Maximum-Likelihood motion estimation of the rest of the links.

Initialization: To initialize the shape, position and orientation of the model we have developed a semi-automatic algorithm whose inputs are a predefined three-dimensional triangular mesh of the human arm, the anthropometric dimensions of the links, and the image position of the joints at time t_0 (Fig. 6). Alternative methods to initialize models from a single video camera are described in [2, 3, 4, 5, 30, 31, 32] and from multiple cameras in [22, 23]. First, the links of the model are scaled according to the real anthropometric dimensions and then their position and orientation is computed from the known image joint positions at time t_0 (assuming that at time t_0 all the links are

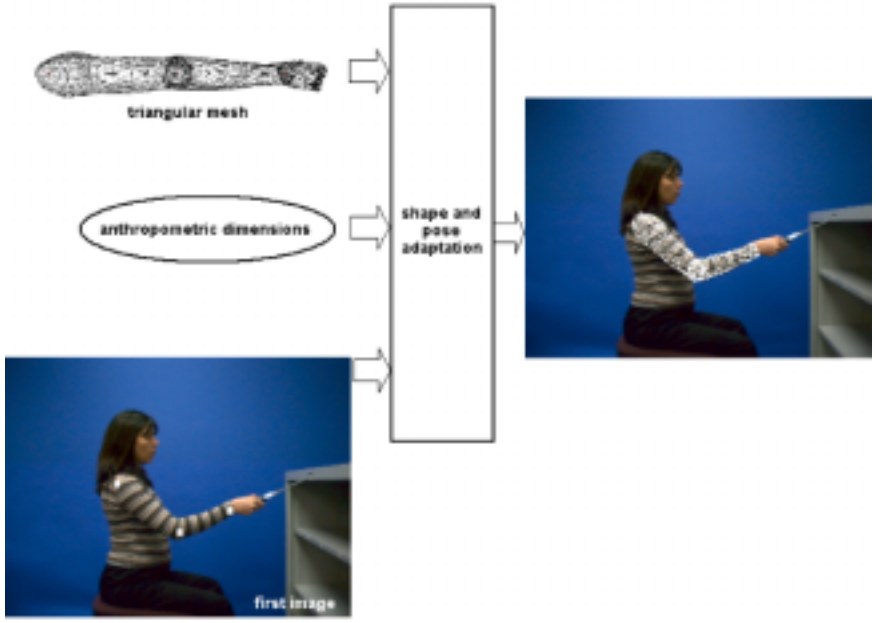


Figure 6: Shape and pose adaptation of an arm model of a subject to the first image of an image sequence.

parallel to the image plane of the camera). Finally, the texture of the articulated model object is obtained by projecting the intensity and chrominance values of the first image of the image sequence to the surface of the model.

Next, points on the surface of the links are selected as observation points (Fig. 7). First the gradient images g_x and g_y are computed by convolving the first intensity image I_0 with the Sobel operator. Then, the vertices of the visible triangles of the links are projected into the camera plane. An image point $\mathbf{a}$ inside the image area of a projected triangle will be selected as an observation point if the linear intensity gradient at position $\mathbf{a}$ satisfies $|\mathbf{g}(\mathbf{a})| > \delta_1$. The corresponding 3D position vector $\mathbf{A}$ is set to the intersection of the observation point's line of sight and the plane containing the vertices of the triangle in 3D. Finally, the intensity value I and the linear intensity gradient of the observation point are set to $I_0(\mathbf{a})$ and $(g_x(\mathbf{a}), g_y(\mathbf{a}))^\top$, respectively. Since the illumination model assumes that the illumination is diffuse as well as spatial and time invariant, the intensity value I and the linear intensity gradient of each observation point remain constant during the image sequence.

Figure 7: Selected observation points for the upper and lower arm of a person. Only surface points with hight linear intensity gradient were selected for motion estimation

Motion estimation of the root: For the Maximum-Likelihood estimation of the parameters $\mathbf{B}_0 = (\Delta T_x^0, \Delta T_y^0, \Delta T_z^0, \Delta\omega_x^0, \Delta\omega_y^0, \Delta\omega_z^0)^\top$ of the root link L_0 only frame to frame intensity difference at observation points of the root link L_0 are evaluated. In order to improve the reliability and accuracy of the estimates the algorithm is applied iteratively [34, 32]. The resulting estimates ${}^i\widehat{\mathbf{B}}_0$ from each iteration i are used to update the motion estimates $\widehat{\mathbf{B}}_0$ found by previous iterations. After each iteration i the root link and its observation points are moved using the estimates ${}^i\widehat{\mathbf{B}}_0$. Due to the motion compensation, the frame to frame intensity differences at the observation points decreases. The iteration ends when after two consecutive iterations the mean square frame to frame intensity difference at the observation does not decrease significantly ($\delta_2 = 10^{-5}$). In each iteration i the following steps are carried out:

1. Compute the covariance matrix ${}^i\mathbf{U}_0$ of the intensity error using Eq. 7.
2. Evaluate Eq. 9 at each observation point.
3. Compute the intensity differences ${}^i\mathbf{FD}_0$ and system matrix ${}^i\mathbf{O}_0$ according to Eq. 14.
4. Estimate the motion estimates ${}^i\widehat{\mathbf{B}}_0$ using Eq. 17.

5. Move the vertices of the mesh of the root link L_0 and its observation points according to Eq. 1 using the estimates ${}^{i}\widehat{\mathbf{B}}_0$.
6. Compute the mean squared intensity difference ${}^{i}msd$.
7. Update the rotation matrix: $\widehat{\Delta\mathbf{R}}_0 \leftarrow {}^{i}\widehat{\Delta\mathbf{R}}_0 \cdot \widehat{\Delta\mathbf{R}}_0$
8. Update the translation vector: $\widehat{\Delta\mathbf{T}}_0 \leftarrow \widehat{\Delta\mathbf{T}}_0 + {}^{i}\widehat{\Delta\mathbf{T}}_0$
9. If $|{}^{i}msd - {}^{i-1}msd| \geq \delta_2$ goto step 1.

Motion estimation of the rest of the links: After the estimation of the motion parameters $\mathbf{B}_0$ of the root link L_0, the rotation angles $\Delta\omega_x^j$, $\Delta\omega_y^j$, and $\Delta\omega_z^j$ of the rest of the links L_j, $j = 1, ..., M-1$, are estimated starting from the root link L_0 one after the other. The motion parameters $\mathbf{B}_j = (\Delta\omega_x^j, \Delta\omega_y^j, \Delta\omega_z^j)^\top$ of link L_j are estimated using only the frame to frame intensity differences at observation points of that link. Before the estimation of the rotation angles $\mathbf{B}_j$ the following steps are carried out:

1. Estimate the translation vector $\Delta\mathbf{T}_j$ using Eq. 2.
2. Translate the vertices of the mesh of the link L_j, its observation points, and the joint J_j according to Eq. 3.

Then, the rotation angles $\mathbf{B}_j$ are estimated by applying an iterative Maximum-Likelihood motion estimation algorithm. In each iteration i the following steps are carried out:

1. Compute the covariance matrix ${}^{i}\mathbf{U}_j$ of the intensity error using Eq. 7.
2. Evaluate $fd(\mathbf{a}) = Q$ (Eq. 10) at each observation point.
3. Compute the intensity differences ${}^{i}\mathbf{FD}_j$ and system matrix ${}^{i}\mathbf{O}_j$ according to Eq. 14.
4. Estimate the rotation angles ${}^{i}\widehat{\mathbf{B}}_j$ using Eq. 17.
5. Rotate the vertices of the mesh of the link L_j and its observation points according to Eq. 4 using the rotation angles ${}^{i}\widehat{\mathbf{B}}_j$.
6. Compute the mean squared intensity difference ${}^{i}msd$.
7. Update the rotation matrix: $\widehat{\Delta\mathbf{R}}_j \leftarrow {}^{i}\widehat{\Delta\mathbf{R}}_j \cdot \widehat{\Delta\mathbf{R}}_j$
8. If $|{}^{i}msd - {}^{i-1}msd| \geq \delta_2$ goto step 1.

3 Commanding the ROBONAUT

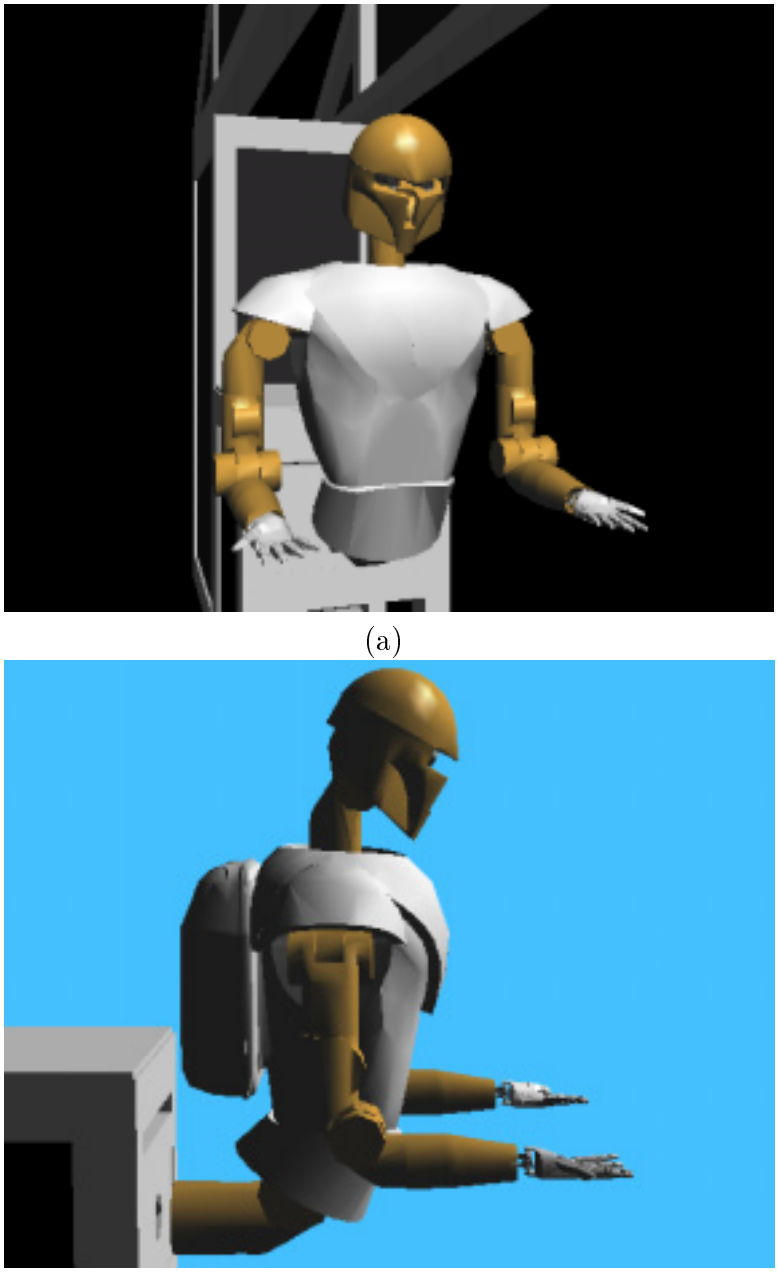
In this section, we present our procedure to remotely command the right arm of the ROBONAUT. Our goal is for the ROBONAUT to imitate the movements of the operator.

Instead of using the real ROBONAUT, we use a ROBONAUT simulation developed at NASA - JSC (Fig. 8) [15]. The ROBONAUT simulation matches the appearance and kinematics of the real ROBONAUT, and it's state is controlled from other processes just like the real ROBONAUT. The control communication is done through a ROBONAUT API developed at NASA - JSC [6]. The ROBONAUT API gives us the ability to see ROBONAUT's sensor data and to command ROBONAUT. The only difference in the interface between ROBONAUT and its simulation is that some sensor data coming from the simulation is not valid. As its underlying communications package, the ROBONAUT API uses the commercial product Real Time Innovations, Inc. (RTI) Network Data Delivery Service (NDDS). The graphics of the ROBONAUT simulation are created by the Enigma Core libraries available from NASA - JSC Interactive Graphics, Operations, and Analysis Laboratory (IGOAL)[14]. From this point onward, ROBONAUT and its simulation will be treated the same.

The variables for the ROBONAUT's right arm that are available through the ROBONAUT API are the position and orientation of the ROBONAUT's right palm expressed in terms of a coordinate system located on the ROBONAUT's chest, the joint angles, and the palm, and joint position limits. The root joint for the inverse kinematics computations is the ROBONAUT's right shoulder position. The palm's end effector is located on the back of the right hand, 1.5 inches forward from the wrist and 0.5 inches towards the back of the hand. The only currently supported method of commanding the ROBONAUT's right arm is sending a message through the ROBONAUT API with a new desired position and orientation for the ROBONAUT's right palm. The arrival of the message on the ROBONAUT site triggers the ROBONAUT's control system, which acts rotating and moving the ROBONAUT's right palm to the new desired position and orientation. The new wrist and elbow positions are computed by using inverse kinematics.

For the teleoperation of the ROBONAUT's right arm, we apply the following 5 steps to each image I_k of the image sequence:

1. Estimate the motion parameters of the operator's right arm from time



(a)

(b)

Figure 8: (a) Coronal, and (b) sagittal view of the virtual ROBONAUT from the simulation developed at NASA - JSC.

t_{k-1} to t_k. For motion estimation apply the Maximum-Likelihood algorithm described in Section 2. Let $\widehat{\Delta \mathbf{T}_0}$, $\widehat{\Delta \mathbf{R}_0}$ be the estimated translation vector and rotation matrix of the operator's right upper arm, respectively, and let $\widehat{\Delta \mathbf{R}_1}$ be the estimated rotation matrix of the operator's right lower arm.

2. Compute the operator's right palm position $\mathbf{J}_2'$ at time t_k:

$$\mathbf{J}_2^{'} = \widehat{\Delta \mathbf{R}_1} \cdot (\mathbf{J}_2 - \mathbf{J}_1) + \mathbf{J}_1 + \widehat{\Delta \mathbf{R}_0} \cdot (\mathbf{J}_1 - \mathbf{J}_0) + \mathbf{J}_0 + \widehat{\Delta \mathbf{T}_0} - \mathbf{J}_1 \ ,$$

where the palm is assumed to be a rigid extension of the lower arm, and $\mathbf{J}_0$, $\mathbf{J}_1$, $\mathbf{J}_2$ are the positions of the operator's right shoulder, elbow and palm at time t_{k-1}, respectively.

3. Compute the translation vector of the operator's right palm from time t_{k-1} to t_k:

$$\Delta \mathbf{J}_2 = \mathbf{J}_2' - \mathbf{J}_2 \ .$$

4. Read the current ROBONAUT's right palm position $\mathbf{J}_{robot}$ through the ROBONAUT API.

5. Compute the new desired ROBONAUT's right palm position $\mathbf{J}'_{robot}$:

$$\mathbf{J}'_{robot} = \mathbf{J}_{robot} + \Delta \mathbf{J}_2 \ .$$

6. Send a message through the ROBONAUT API with the new desired ROBONAUT's right palm position $\mathbf{J}'_{robot}$.

4 Experimental Results

We have implemented the Maximum-Likelihood motion estimation algorithm described under Windows 2000, and have performed a number of experiments on synthetic and real image sequences to assess its accuracy, limitations, and advantages for estimating the motion of the right arm of a person. The real image sequences were obtained using a Pulnix TMC-9700 1-2/3"CCD

Progressive Scan Color Video Camera with a 2/3" 9 mm lens and a 640x480 RGB video output at a frame rate of 30 Hz. The video signal was acquired using a Matrox Meteor-II/Multi-Channel frame grabber. All the experiments were performed on a Pentium III (1Gz) workstation with 0.5GB RAM. The average processing time was 1.02 s per frame. The minimum and maximum processing time was 0.43 s and 8.65 s per frame, respectively. In all the experiments, the thresholds δ_1 and δ_2 were set to 20 and 10^{-5}, respectively. These values were experimentally determined. We present the experimental results obtained from two synthetic image sequences called HAZEL-S as well as three real image sequences called HAZEL-A, HAZEL-B and HAZEL-C.

For the first experiment, we applied the Maximum-Likelihood motion estimation algorithm to the synthetic image sequence HAZEL-S depicting a moving right arm from a virtual human. The dimensions of the arm correspond to the dimensions of the right arm of one of the co-authors. The synthetic image sequence was generated by obtaining 400 images (RGB, 640x480 pixels^2) of the arm at different times while the arm is moving along a predefined trajectory. Fig. 9 depicts six frames from the synthetic image sequence. The maximum value of the magnitude of the frame to frame translation vector of the shoulder, elbow, and wrist in the 3D virtual world is 0.5 cm, 1.15 cm, 1.89 cm, respectively. The maximum value of the magnitude of the frame to frame displacement vector of the shoulder, elbow, and wrist in the image plane is 2.22 pixels, 4.95 pixels, and 9.15 pixels, respectively.

Fig. 10 depicts the estimated translation vector and the rotation angles of the upper arm, and the estimated rotation angles of the lower arm from each frame of the synthetic image sequence to the next. Fig. 11 depicts the ground truth and the positions (computed using the estimated motion parameters) of the shoulder, the elbow, and wrist for the first 100 frames of the synthetic image sequence. Fig. 12 depicts the magnitude of the position error for the shoulder, the elbow, and the wrist for all the frames of the synthetic image sequence. The mean of the magnitude of the position error of the shoulder, the elbow and the wrist is as follows: $|\Delta\mathbf{A}_{shoulder}| = 0.056743$ cm, $|\Delta\mathbf{A}_{elbow}| = 0.049568$ cm, and $|\Delta\mathbf{A}_{wrist}| = 0.055171$ cm, while the variance is 0.003976 cm^2, 0.002582 cm^2,and 0.009226 cm^2 respectively. According to Fig. 12 the magnitude of the position error appears to increase quickly at the beginning of the image sequence. However, later in the image sequence the magnitude of the position error stops to increase and begins to decrease. The plots indicate that the motion estimation algorithm is able to recover from previous position errors. This is due to the fact that previous position

errors are being taken into account for motion estimation by means of the covariance matrix.

For the second experiment, we tested the Maximum-Likelihood motion estimation algorithm using the HAZEL-C sequence (354 frames) depicting a woman moving her right index finger along a rectangle with known position, orientation and dimensions (Figs. 13(a-f)). Figs. 13(g-l) depict the model at the estimated position and orientation overlayed at the image sequence. Fig. 14 depicts the estimated translation vector and the rotation angles of the upper arm and the estimated rotation angles of the lower arm from each frame of the real image sequence HAZEL-C to the next. Fig. 15 depicts the ground truth and the positions (computed using the estimated motion parameters) of the right index finger for the subject depicted in the HAZEL-C sequence. Figs. 16(a-c) depict the mean of the absolute position error along the x, y and z axis of the world coordinate system for all the frames of the image sequence. Fig. 16(d) depicts the mean of the magnitude of the position error for all the frames of the image sequence. The mean of the magnitude of the position error of the right index finger is $|\Delta \mathbf{A}_{finger}| = 0.570256$ cm, while the variance is 0.099233 cm^2. The mean and variance of the magnitude of the position error on the image plane is 1.099577 pixel and 0.586255 pixel^2, respectively. The minimum and the maximum value of the magnitude of the position error is 0.062808 cm and 1.856693 cm, respectively. The minimum and maximum value of the magnitude of the position error on the image plane is 0.043543 pixel and 3.142040 pixel, respectively. The minimum and maximum value of the magnitude of the component of the position error parallel to the image plane is 0.024343 cm and 1.711261 cm, respectively. The minimum and maximum value of magnitude of the component of the position error perpendicular to the image plane is 0.001584 cm and 1.723885 cm, respectively.

For the third experiment, we tested the Maximum-Likelihood Motion Estimation Algorithm using the HAZEL-A and HAZEL-B sequences (200 frames each) depicting a woman grasping (Figs. 17(a-f)) and moving (Figs. 19(a-f)) an object in front of a bookshelf. Figs. 18 and 20 depict the estimated translation vector and rotation angles of the upper arm and the estimated rotation angles of the lower arm from each frame of the real image sequence HAZEL-A to the next and from each frame of the image sequence and HAZEL-B to the next, respectively. Figs. 17(g-l) and 19(g-l) depict the models at the estimated position and orientation overlayed at the image sequences. Although the model remains well aligned during tracking, some

position errors can still be observed. For example, a peak error is observed in all the curves in Fig. 18 for frame 116. In that frame (Fig. 17(j)) the shoulder has drifted backwards. However, the algorithm quickly compensated for these errors and tracking was not lost.

Figs. 21(a-f) and Figs. 21(g-l) depict the coronal and the sagittal view of the virtual ROBONAUT from the NASA being animated with the estimated motion parameters of HAZEL-A. Figs. 22(a-f) and Figs. 22(g-l) depict the coronal and the sagittal view of the virtual ROBONAUT being animated with the estimated motion parameters of HAZEL-B.

5 Conclusions

We have implemented the Maximum-Likelihood motion estimation algorithm of articulated objects proposed in [34, 32] and applied it for estimating the motion of a moving human arm. Then, we performed a number of experiments on synthetic and real data to assess its accuracy, limitations and advantages. The experimental results with synthetic image sequences revealed a position error for the shoulder, elbow and wrist of 0.6 ± 0.6 mm, 0.5 ± 0.5 mm and 0.6 ± 1.0 mm, respectively. The experimental results with real image sequences revealed a position error for the right index finger of 0.57 ± 0.31 cm. Furthermore, the model object remained well aligned during the tested image sequences. Although some position errors could be observed, the algorithm was able to compensate for those errors and never lost tracking. This ability for recovering from previous position errors is due to the fact that previous position errors are being taken into account during motion estimation. Finally, we have used the motion estimates to remotely command the right arm of a virtual ROBONAUT. The control communication is done through a ROBONAUT API developed at NASA - JSC and the commercial product Real Time Innovations, Inc. (RTI) Network Data Delivery Service (NDDS).

Acknowledgments

We acknowledge the support of the University of Houston's Institute for Space Systems Operations (http://www.isso.uh.edu) with a Postdoctoral Fellowship to Dr. Martinez and the support of Real Time Innovations, Inc., with a software grant (NDDS).

References

[1] J.K. Aggarwal and Q. Cai. Human motion analysis: A Review. *Computer Vision and Image Understanding*, 73(3), 1999.

[2] C. Barrón and I.A. Kakadiaris. Estimating anthropometry and pose from a single image. In *Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 669–676, Hilton Head Island, SC, June 13-15 2000.

[3] C. Barrón and I.A. Kakadiaris. On the improvement of anthropometry and pose estimation from a single uncalibrated image. In *IEEE Workshop on Human Motion*, pages 53–60, Austin, TX, December 7-8 2000.

[4] C. Barrón and I.A. Kakadiaris. Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding*, 81(3):269–284, 2001.

[5] C. Barrón and I.A. Kakadiaris. On the improvement of anthropometry and pose estimation from a single uncalibrated image. Submitted to the *Machine Vision and Applications - Special Issue on Human Modeling, Analysis and Synthesis*, May 2002.

[6] B. Bluethmann. ROBONAUT API: Version 1.0. Manual, Dexterous Robotics Laboratory, NASA - Johnson Space Center, 2001.

[7] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8–15, Santa Barbara, CA, June 23-25 1998.

[8] Z. Chen and H.J. Lee. Knowledge-guided visual perception of 3D human gait from single image sequence. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):336 – 342, 1992.

[9] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal in Computer Vision*, 38(2):99–127, July 2000.

[10] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *Proceedings of the 7th International Conference on Computer Vision*, pages 716–721, Kerkyra, Greece, September 20-27 1999.

[11] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1), January 1999.

[12] D.M. Gavrila and L.S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *Proceedings of the 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, CA, June 18-20 1996.

[13] Luis Goncalves, Enrico Di Bernardom, Enrico Ursella, and Pietro Perona. Monocular tracking of the human arm in 3D. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 764–770, Boston, MA, June 20-22 1995.

[14] M. Goza. Enigma user manual. Manual, Interactive Graphics, Operations, and Analysis Laboratory (IGOAL), NASA - Johnson Space Center, 2001.

[15] M. Goza. ROBONAUT API: Robosim v2.2. Manual, Dexterous Robotics Laboratory, NASA - Johnson Space Center, 2001.

[16] D. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5 – 20, 1983.

[17] R.J. Holt, A.N. Netravali, T. S. Huang, and R.J. Qian. Determining articulated motion from perspective views: A decomposition approach. *Pattern Recognition*, 30:1435–1449, 1997.

[18] Y. Iwai, K. Ogaki, and M. Yachida. Posture estimation using structure and motion models. In *Proceedings of the International Conference on Computer Vision*, pages 214 – 219, 1999.

[19] S.X. Ju, M.J. Black, and Y. Yacoob. Cardboard: a parameterized model of articulated image motion. In *Proceedings of the Second International Workshop on Automatic Face and Gesture Recognition*, pages 38 – 44, September 1996.

[20] I. A. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 81–87, San Francisco, CA, June 18–20 1996.

[21] I.A. Kakadiaris. Optical tracking for telepresence and teleoperation space applications. White paper, University of Houston, Houston, TX, 1999.

[22] I.A. Kakadiaris and D. Metaxas. 3D human body model acquisition from multiple views. In *Proceedings of the International Conference on Computer Vision*, pages 618–623, Boston, MA, June 20-23 1995.

[23] I.A. Kakadiaris and D. Metaxas. 3D human body model acquisition from multiple views. *International Journal of Computer Vision*, 30(3):191–218, 1998.

[24] I.A. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1453–1459, 2000.

[25] I.A. Kakadiaris and R. Sharma, editors. *Proceedings of the IEEE Human Motion Analysis and Synthesis Workshop*, Hilton Head Island, South Carolina, June 2000. IEEE Computer Society Press.

[26] I.A. Kakadiaris, R. Sharma, and M. Yeasin. Editorial comments on the special issue on human modeling, analysis and synthesis. *Machine Vision and Applications*, May 2002.

[27] F. Kappei and C. Liedtke. Modelling of a 3-D scene consisting of moving objects from a sequence of monocular TV images. In *Proceedings of the Real Time Image Processing SPIE Conference: Concepts and Technologies*, pages 126–130, 1987.

[28] R. Koch. Dynamic 3D scene analysis through synthesis feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):556–568, June 1993.

[29] H. Li, P. Roivainen, and R. Forchheimer. 3D motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):545–555, June 1993.

[30] G. Martinez. Shape estimation of articulated 3D objects considering mutual occlusions for object-based analysis-synthesis coding (OBASC). In *Proceedings of the Picture Coding Symposium*, Melbourne, Australia, March 1996.

[31] G. Martinez. Shape estimation of articulated objects for object-based analysis-synthesis coding (obasc). *Signal Processing: Image Communications*, 9(3):213 – 216, 1997.

[32] G. Martinez. *Analyse-Synthese-Codierung basierend auf dem Modell bewegter dreidimensionaler, gegliederter Objecte*. PhD thesis, University of Hannover, Hannover, Germany, 1998.

[33] G. Martinez. Analysis-synthesis coding based on the source model of articulated three-dimensional objects. In *Proceedings of the Picture Coding Symposium*, Portland, Oregon, March 1999.

[34] G. Martinez. Maximum-likelihood motion estimation of articulated objects for object-based analysis-synthesis coding. In *Proceedings of the Picture Coding Symposium 2001*, pages 293 – 396, Seul, Korea, April 25-27 2001.

[35] D. Meyer, J. Denzler, and H. Niemann. Model based extraction of articulated objects in image sequences. In *Proceedings of the Fourth International Conference on Image Precessing*, pages 78 – 81, 1997.

[36] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *IEEE Transactions on Systems, Man, and Cybernetics*, 81(3):231 – 268, 2001.

[37] Thomas B. Moeslund and Erik Granum. 3D human pose estimation using 2D-data and an alternative phase space representation. In I.A. Kakadiaris and R. Sharma, editors, *Proceedings of the IEEE Workshop on Human Modeling, Analysis and Synthesis*, pages 26–33, Hilton Head Island, SC, June 16 2000.

[38] E-J. Ong and S. Gong. Tracking hybrid 2D-3D human models through multiple views. In *Proceedings of the Workshop on Modelling People at ICCV'99*, pages 11 – 18, September 1999.

[39] J. Ostermann. Object-based analysis-synthesis coding based on the source model of moving rigid 3D objects. *Signal Processing: Image Communications*, 6(2):143–161, 1994.

[40] R. Plankers, P. Fua, and N. D'Apuzzo. Automated body modeling from video sequences. In *Proceedings of the IEEE International Workshop on Modeling People*, pages 45–52, Corfu, Greece, September 20 1999.

[41] K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics and Image Processing*, 59(1):94 – 115, January 1994.

[42] R. Rosales and S. Sclaroff. Learning and synthesizing human body motion and posture. In *Proceedings of the fourth International Conference on Automatic Face and Gesture Recognition*, pages 506 – 511, March 2000.

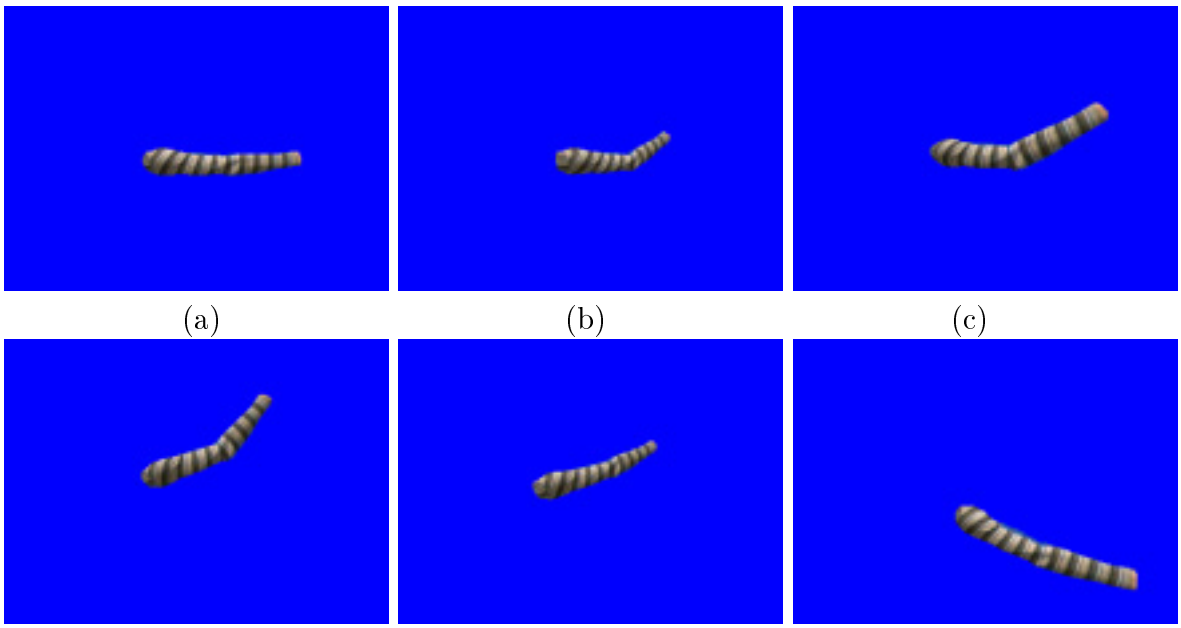
[43] S. Wachter and H.-H. Nagel. Tracking of persons in monocular image sequences. In *Proceedings of IEEE Nonrigid and Articulated Motion Workshop*, pages 2–9, Puerto Rico, June 16 1997. IEEE Computer Society.

[44] C. R. Wren and A. P. Pentland. Dynamic models of human motion. In *Proceedings of the 3rd International Conference on Automatic Face and Gesture Recognition*, pages 22 – 27, Nara, Japan, April 1998.

[45] M. Yamamoto and K. Koshikawa. Human motion analysis based on a robot arm model. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 664 – 665, Lahaina, Maui, Hawaii, June 3 - 6 1991.

[46] M. Yamamoto, Y. Ohta, T. Yamagiwa, and K. Yagishita. Human action tracking guided by key-frames. In *Proceedings of the fourth International Conference on Automatic Face and Gesture Recognition*, pages 354 – 361, March 2000.

[47] M. Yamamoto, A. Sato, and S. Kamada. Incremental tracking of human actions from multiple views. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2 – 7, June 1998.



(a) (b) (c)

(d) (e) (f)

Figure 9: (a-f) Frames 1, 25, 75, 230, 320, and 360 from the synthetic image sequence, respectively.

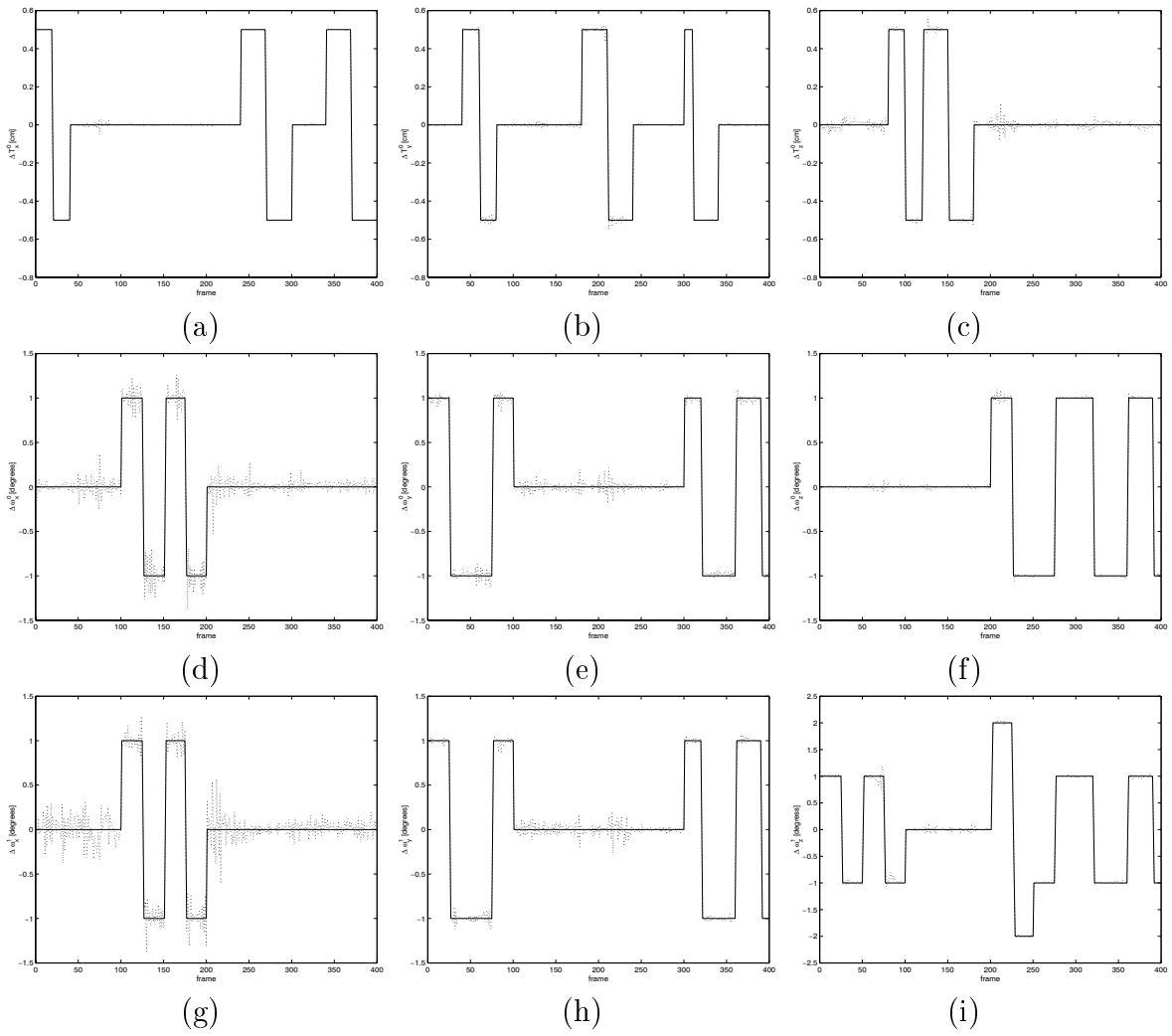
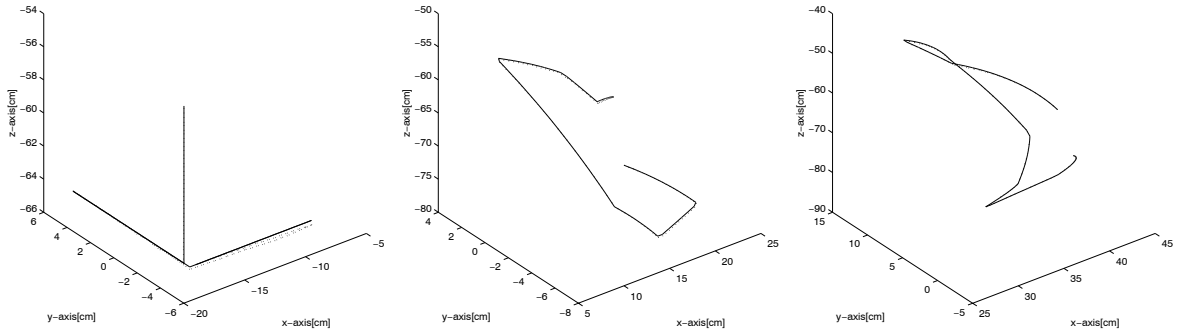
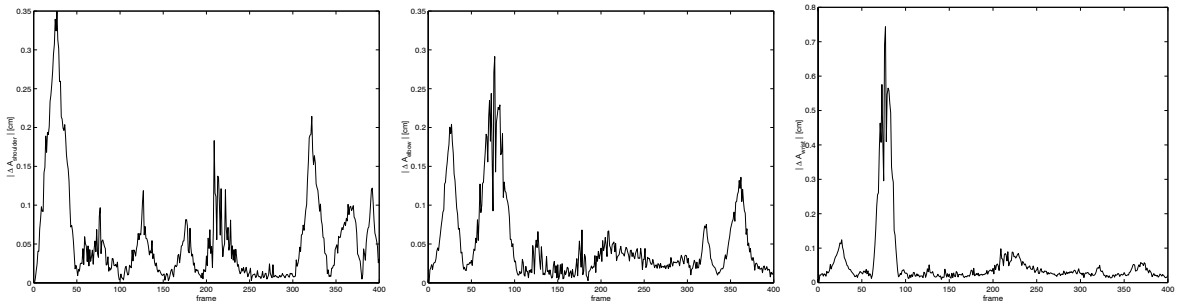


Figure 10: Motion estimation for the moving arm depicted in HAZEL-S1. (a-c) depict the estimated components of the translation vector of the upper arm ($\widehat{\Delta T_x^0}$, $\widehat{\Delta T_y^0}$, and $\widehat{\Delta T_z^0}$), (d-f) depict the estimated rotation angles of the upper arm ($\widehat{\Delta\omega_x^0}$, $\widehat{\Delta\omega_y^0}$, and $\widehat{\Delta\omega_z^0}$), and (g-i) depict the estimated rotation angles of the lower arm ($\widehat{\Delta\omega_x^1}$, $\widehat{\Delta\omega_y^1}$, and $\widehat{\Delta\omega_z^1}$). The solid line represents the ground truth values while the dotted one depicts the estimated values.



(a) (b) (c)

Figure 11: (a-c) Plots of the position of the shoulder, the elbow, and the wrist during the first 100 frames of the image sequence HAZEL-S. The solid line represents the ground truth position and the dotted one the estimated position.



(a) (b) (c)

Figure 12: (a-c) Plots of the magnitude of the position error for the shoulder ($|\Delta\mathbf{A}_{shoulder}|$), the elbow ($|\Delta\mathbf{A}_{elbow}|$), and the wrist ($|\Delta\mathbf{A}_{wrist}|$) of the moving arm for all the frames of the image sequence HAZEL-S.

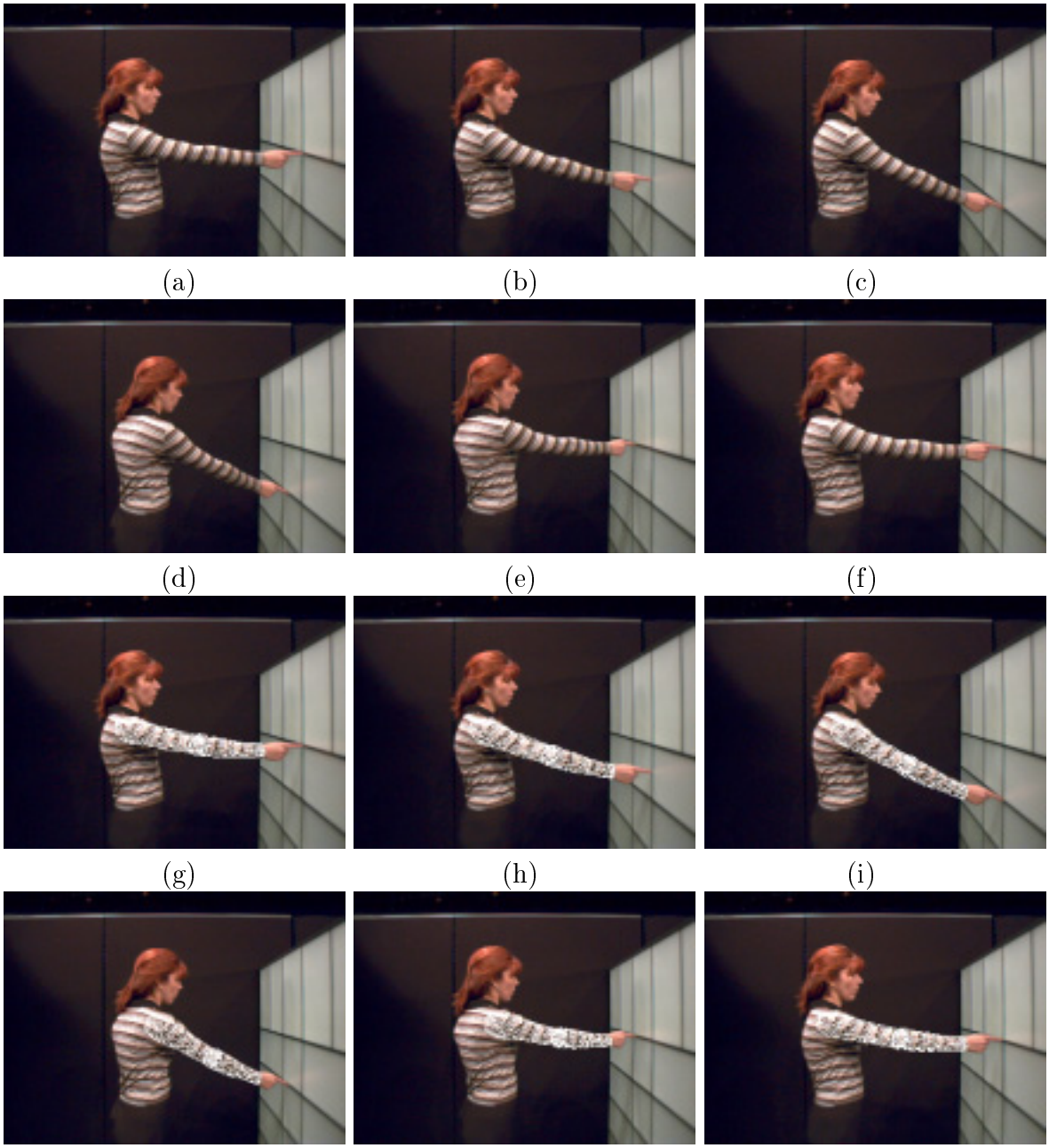


Figure 13: (a-f) Frames 1, 60, 120, 180, 300, and 354 from the sequence HAZEL-C, respectively. (g-l) Original frames with the model overlayed at the estimated position and orientation.

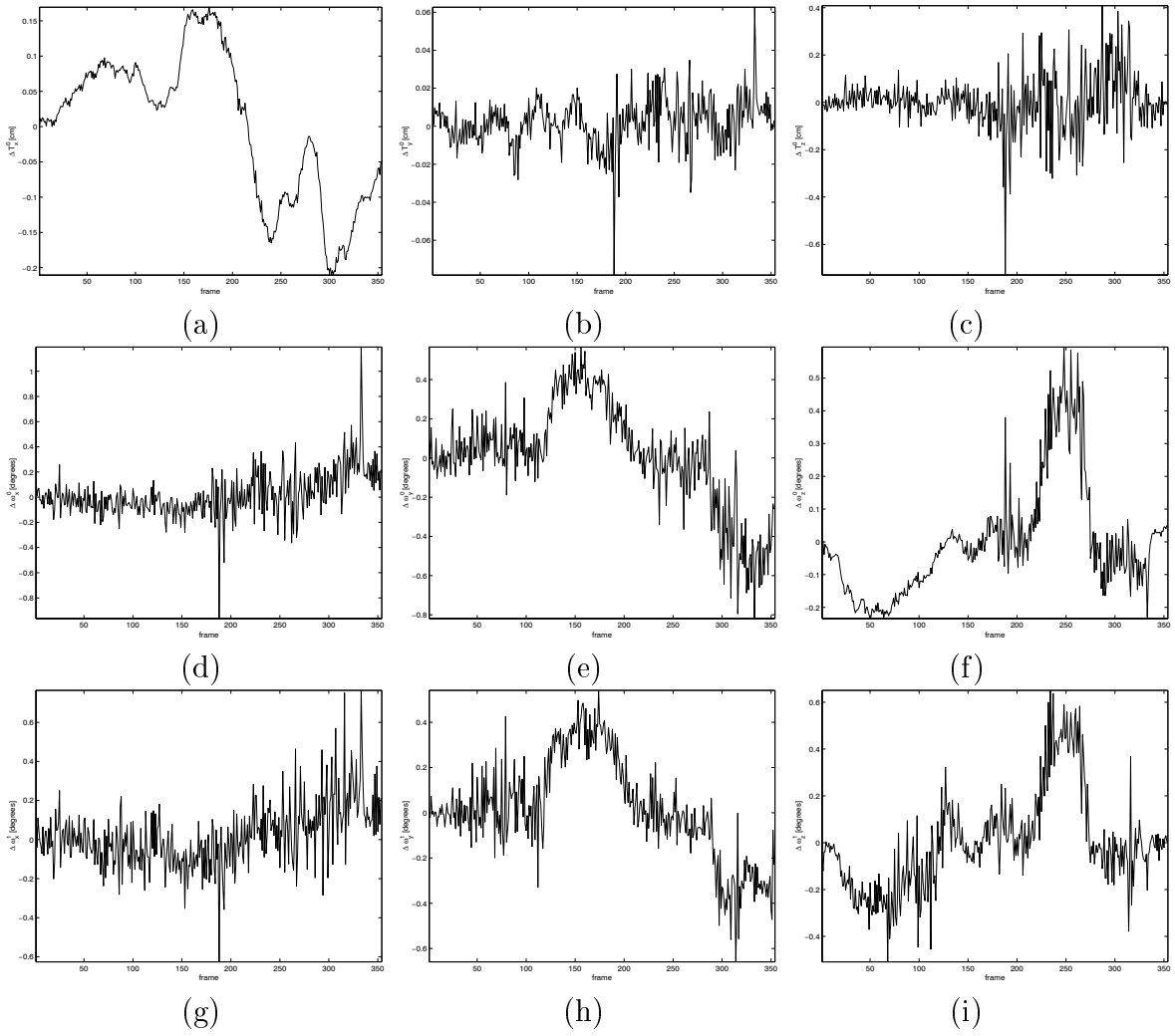
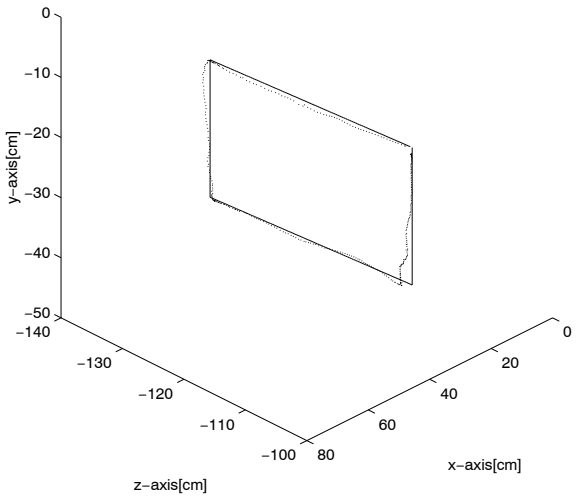
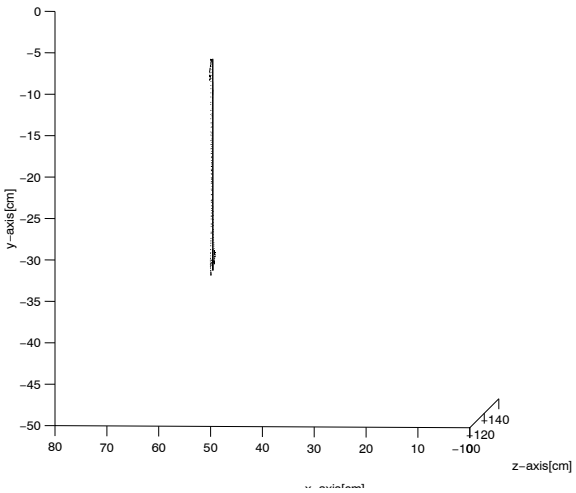


Figure 14: Motion estimation for the moving arm depicted in the image sequence HAZEL-C. (a-c) depict the estimated components of the translation vector of the upper arm ($\widehat{\Delta T_x^0}$, $\widehat{\Delta T_y^0}$, and $\widehat{\Delta T_z^0}$), (d-f) depict the estimated rotation angles of the upper arm ($\widehat{\Delta\omega_x^0}$, $\widehat{\Delta\omega_y^0}$, and $\widehat{\Delta\omega_z^0}$), and (g-i) depict the estimated rotation angles of the lower arm ($\widehat{\Delta\omega_x^1}$, $\widehat{\Delta\omega_y^1}$, and $\widehat{\Delta\omega_y^1}$).

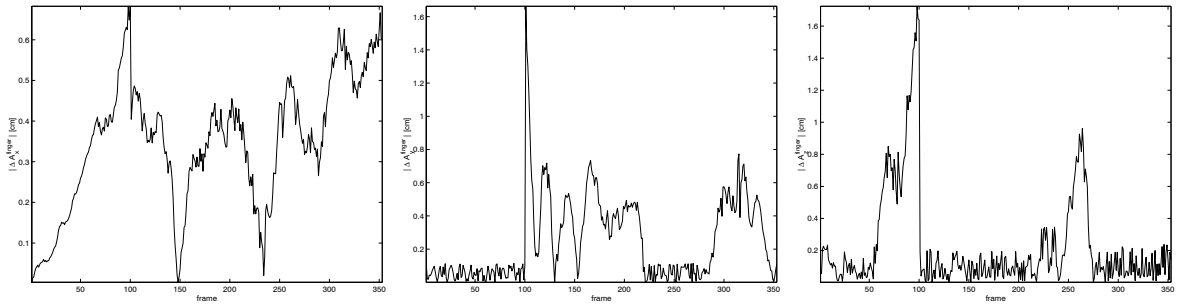


(a)

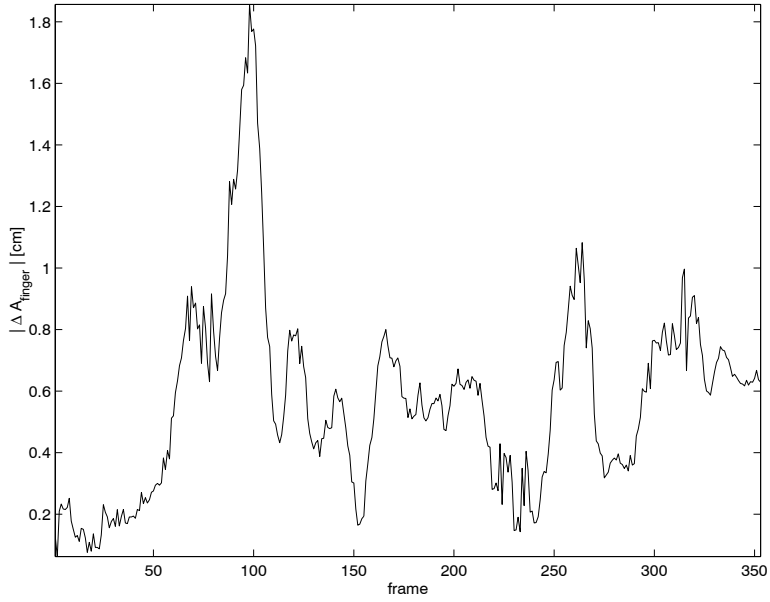


(b)

Figure 15: (a-b) Plots of the position of the right index finger of the subject depicted in the image sequence HAZEL-C. The solid line represents the ground truth position and the dotted one the computed position (using the motion estimates).



(a) (b) (c)



(d)

Figure 16: (a-c) Plots of the absolute position error of the index finger along the x, y, and z axis ($\left|\Delta A_x^{finger}\right|$, $\left|\Delta A_y^{finger}\right|$, $\left|\Delta A_z^{finger}\right|$) for all the frames of the image sequence HAZEL-C. (d) Plot of the magnitude of the position error of the index finger ($|\Delta \mathbf{A}_{finger}|$) for all the frames of the image sequence HAZEL-C.

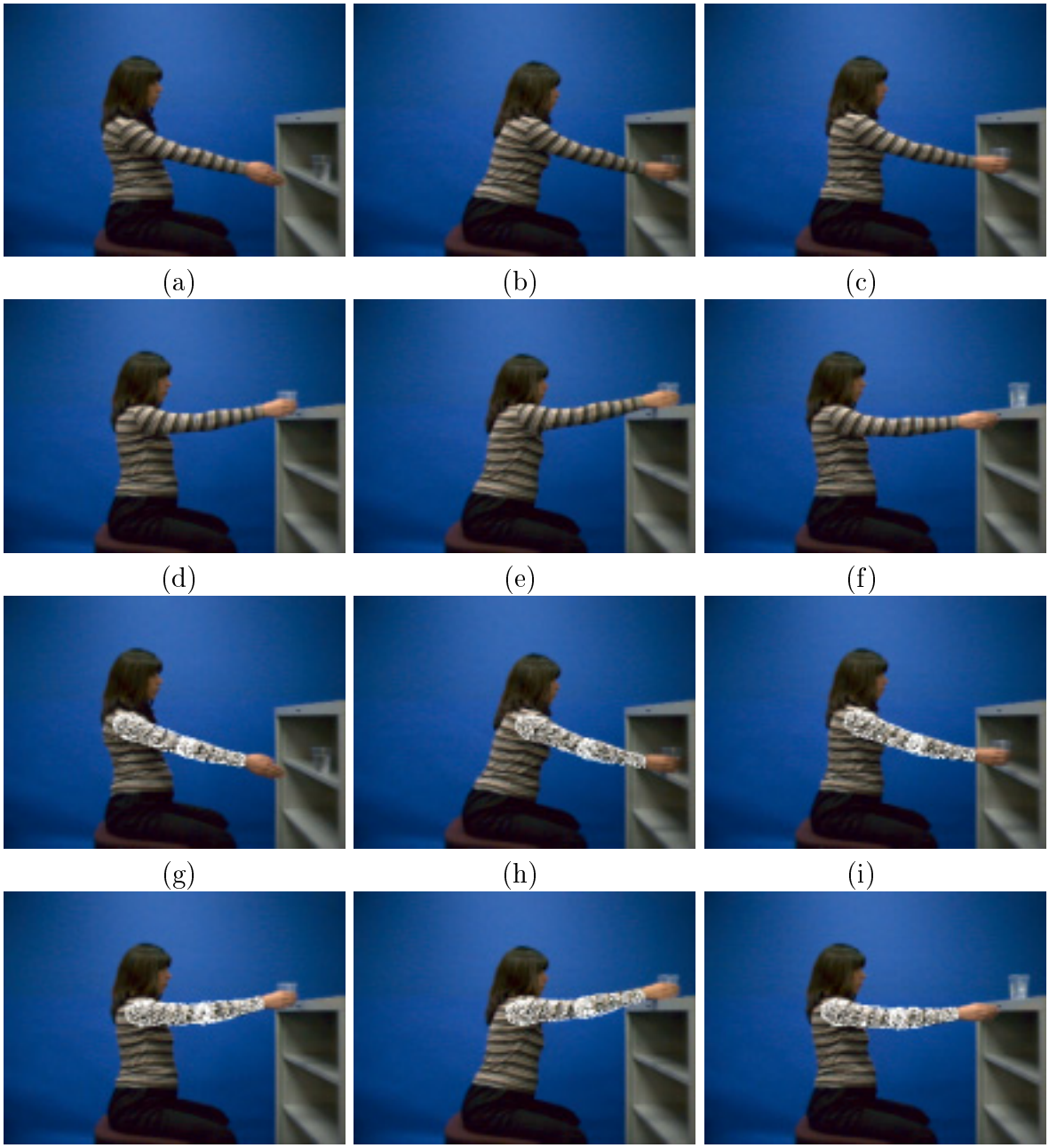


Figure 17: (a-f) Frames 1, 40, 80, 116, 160, and 200 from the sequence HAZEL-A. (g-l) Original frames with the model overlayed at the estimated position and orientation.

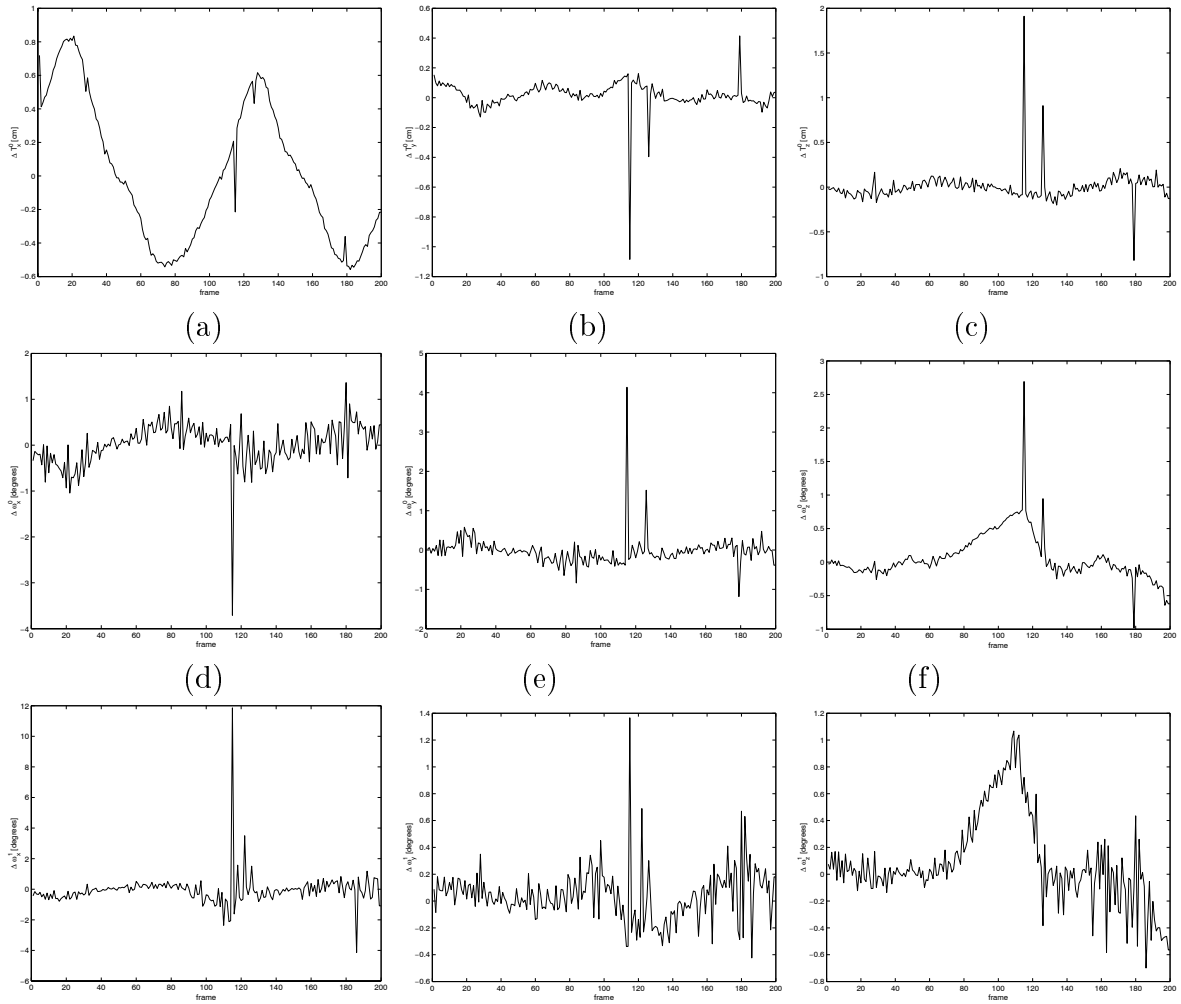


Figure 18: Motion estimation for the moving arm depicted in the image sequence HAZEL-A. (a-c) depict the estimated components of the translation vector of the upper arm ($\widehat{\Delta T_x^0}$, $\widehat{\Delta T_y^0}$, and $\widehat{\Delta T_z^0}$), (d-f) depict the estimated rotation angles of the upper arm ($\widehat{\Delta\omega_x^0}$, $\widehat{\Delta\omega_y^0}$, and $\widehat{\Delta\omega_z^0}$), and (g-i) depict the estimated rotation angles of the lower arm ($\widehat{\Delta\omega_x^1}$, $\widehat{\Delta\omega_y^1}$, and $\widehat{\Delta\omega_y^1}$).

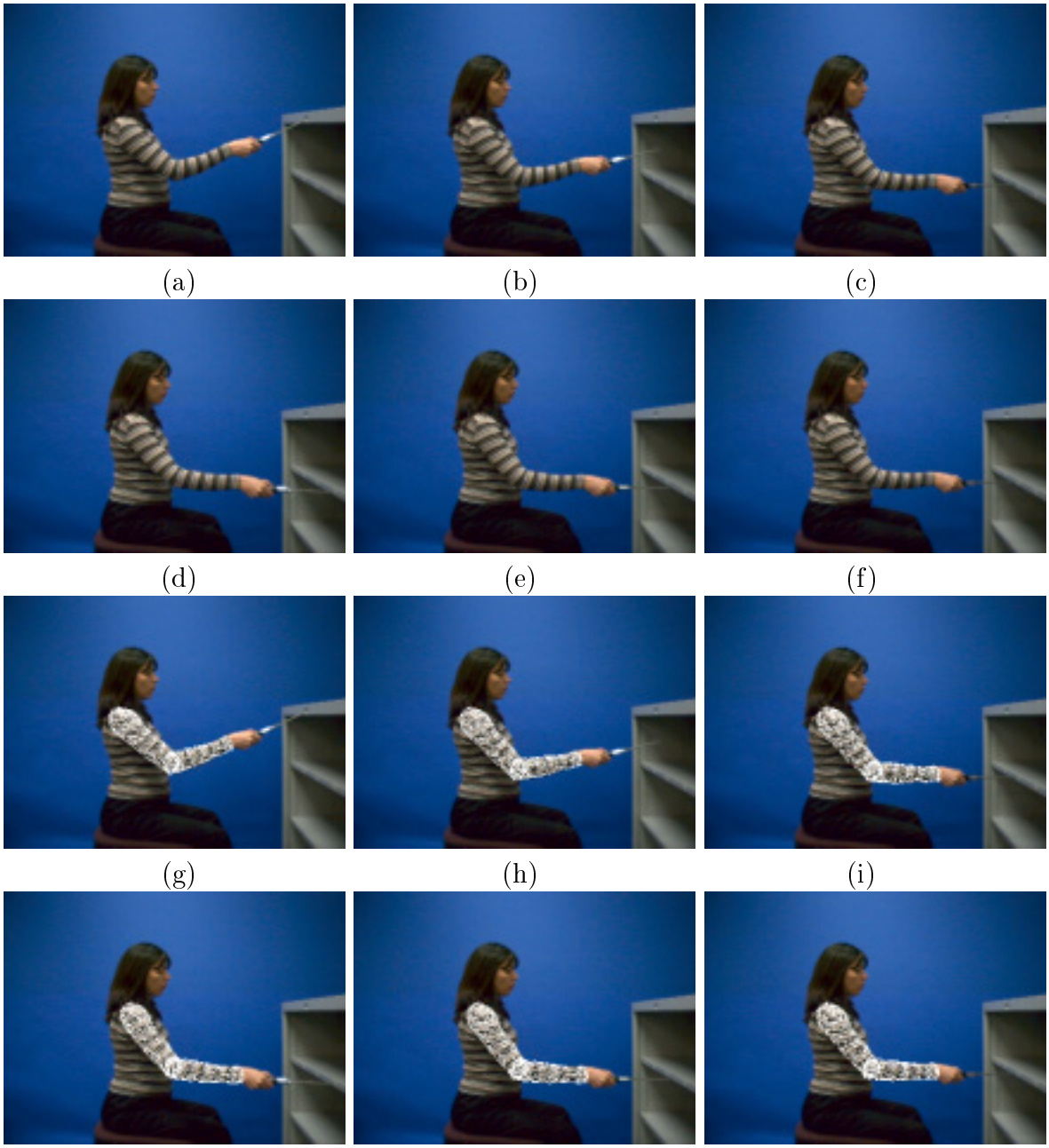


Figure 19: (a-f) Frames 1, 45, 90, 160, 180, and 200 from the sequence HAZEL-B. (g-l) Original frames with the model overlayed at the estimated position and orientation.

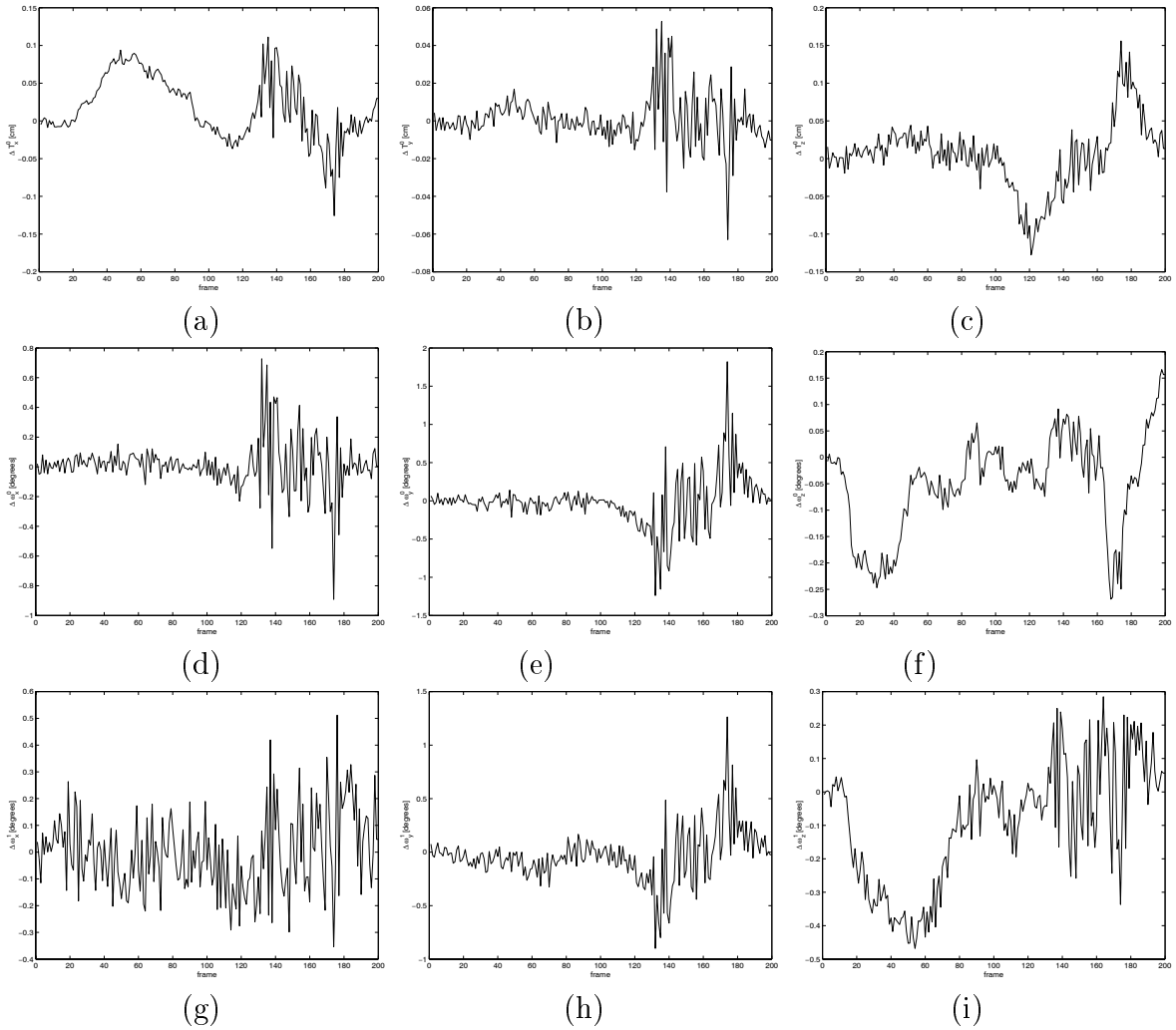


Figure 20: Motion estimation for the moving arm depicted in the image sequence HAZEL-B. (a-c) depict the estimated components of the translation vector of the upper arm ($\widehat{\Delta T_x^0}$, $\widehat{\Delta T_y^0}$, and $\widehat{\Delta T_z^0}$), (d-f) depict the estimated rotation angles of the upper arm ($\widehat{\Delta\omega_x^0}$, $\widehat{\Delta\omega_y^0}$, and $\widehat{\Delta\omega_z^0}$), and (g-i) depict the estimated rotation angles of the lower arm ($\widehat{\Delta\omega_x^1}$, $\widehat{\Delta\omega_y^1}$, and $\widehat{\Delta\omega_y^1}$).

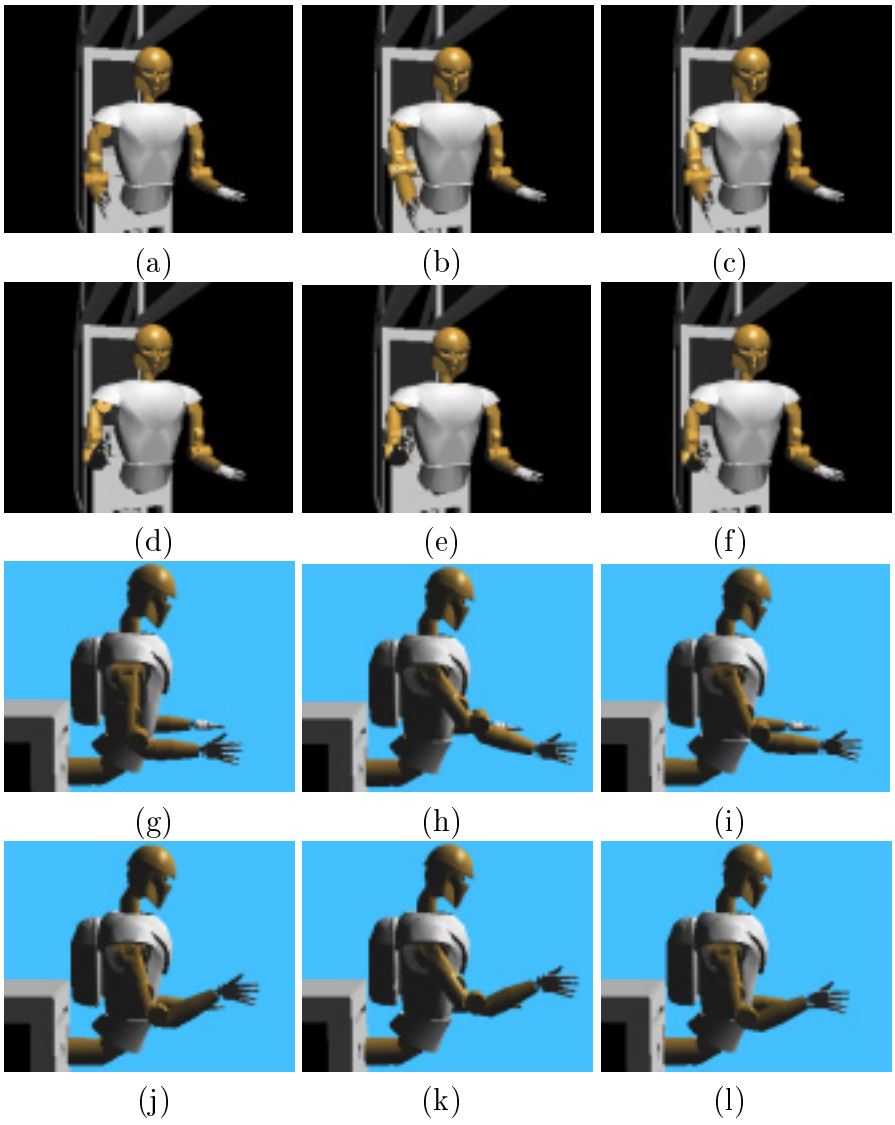


Figure 21: Commanding a ROBONAUT simulation developed at NASA - JSC with the estimated motion parameters of the HAZEL-A sequence. (a-f) Coronal and (g-l) sagittal view of the postures corresponding to the frames 1, 40, 80, 116, 160, and 200 of the sequence.

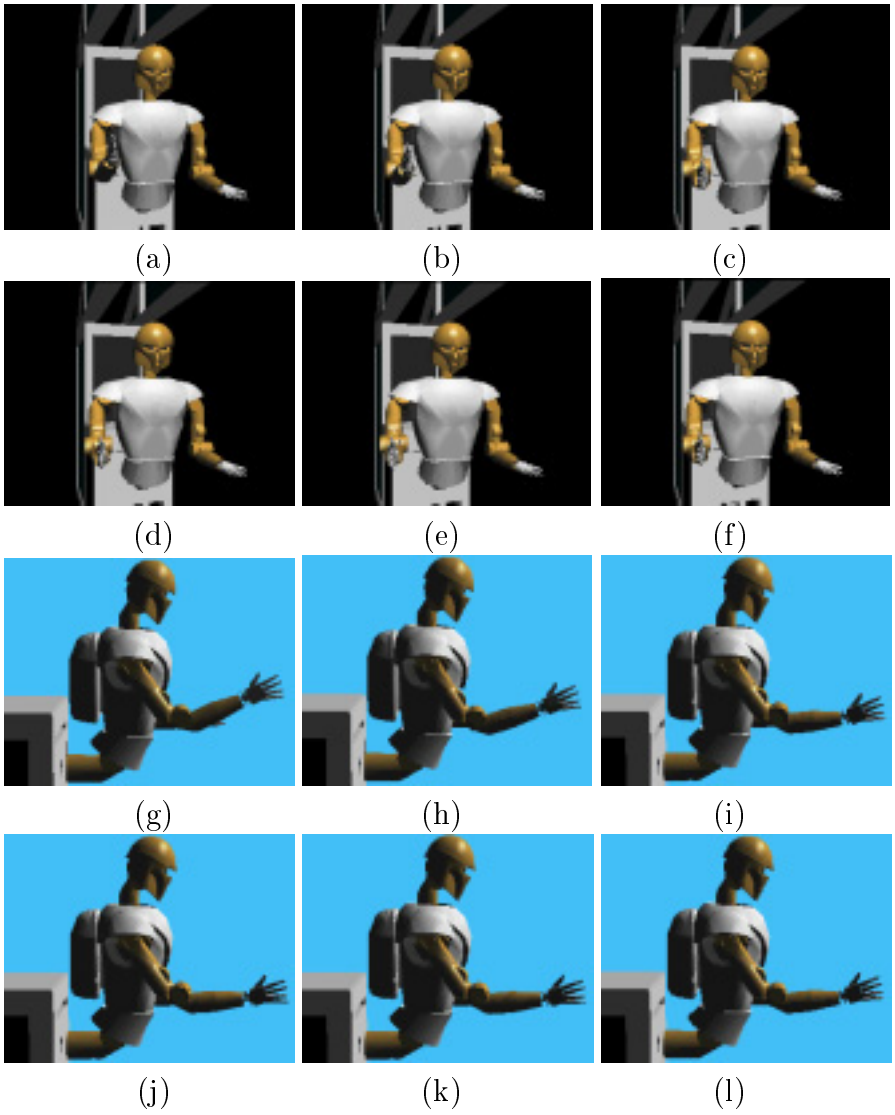


Figure 22: Commanding a ROBONAUT simulation developed at NASA - JSC with the estimated motion parameters of the HAZEL-B sequence. (a-f) Coronal and (g-l) sagittal view of the postures corresponding to the frames 1, 45, 90, 160, 180, and 200 of the sequence.