



Teleoperating ROBONAUT: A case study

G. Martinez, I. A. Kakadiaris

Visual Computing Lab

Department of Computer Science

University of Houston

Houston TX 77204-3010

Email: {geovanni,ioannisk}@uh.edu

D. Magruder

Robotic Systems Technology Branch

Automation and Robotics Division

NASA - Johnson Space Center

Houston, TX-77058

Email: darby.f.magruder1@jsc.nasa.gov

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 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 0.57 ± 0.31 cm. The motion estimates were used to remotely command a robonaut simulation developed at NASA - JSC.

1 Introduction

Human exploration and development of space will demand a heavy extravehicular activity (EVA) workload (space walks) from a small number of crewmembers. In order to alleviate the astronaut workload robots remotely working with teleoperated control are currently being developed at NASA - Johnson Space Center. One such robot is the ROBONAUT (ROBOtic astroNAUT), which is an anthropomorphic robot with two arms, two hands, a head, a torso, and a stabilizing leg [2]. The arms are capable of dexterous, human-like maneuvers to handle common EVA tools.

One intuitive way to teleoperate the ROBONAUT is to estimate the three-dimensional motion of the teleoperator's body parts (e.g., head, arms, torso, and legs) and then use the estimated motion parameters to control the ROBONAUT [12]. In such a system, the robot duplicates the movements made by a teleoperator. As the teleoperator extends out his/her arm, so does the ROBONAUT. And if the teleoperator starts twisting a screwdriver, the ROBONAUT should duplicate the action.

The existing literature on human motion estimation from a monocular image sequence can be roughly divided into two groups (see [1, 6, 16] for comprehensive reviews). The



first one estimates the motion using image features (e.g. edge points) [7, 8, 11]. The second group estimates the motion from frame to frame intensity differences at observation points [4, 13, 17]. 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 [5] both image features and frame to frame intensity differences are taken into account for motion estimation. In this contribution, the motion is estimated by maximizing the conditional probability of the frame to frame intensity difference at observation points [14, 15].

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 [10]. Finally, we perform a number of experiments on 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 real image sequences are detailed. Finally, in Section 5, we offer our conclusions.

2 Maximum-Likelihood Motion Estimation

Figure 1 depicts an articulated object consisting of two rigid links L_0 and L_1 connected by a spherical joint J_1 (at times t_k and t_{k+1}). In the following, without loss of generality, we assume that the link L_0 is the root link of the articulated object and that the shape, position, and orientation of both links as well as the position of the joint $\mathbf{J}_1 = (J_x^{(1)}, J_y^{(1)}, J_z^{(1)})^T$ at time t_k are known or previously estimated. In addition, the origin of the coordinate system of the link L_1 is placed at the joint position \mathbf{J}_1 and the origin of the coordinate system of the root link is placed at an arbitrary point \mathbf{J}_0 inside its mesh.

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. The translation is described by the 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:

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

The motion of the link L_1 from discrete time t_k to t_{k+1} is described first by a translation and then by a rotation of its local coordinate system in the world coordinate system. The rotation is described by the rotation matrix $\Delta\mathbf{R}_1$ defined by the three rotation angles $\Delta\omega_x^{(1)}$, $\Delta\omega_y^{(1)}$, and $\Delta\omega_z^{(1)}$. The translation is described by the 3D translation vector $\Delta\mathbf{T}_1 = (\Delta T_x^{(1)}, \Delta T_y^{(1)}, \Delta T_z^{(1)})$. Due to the constraints imposed by the joint \mathbf{J}_1 on the motion of the link L_1 , the translation vector $\Delta\mathbf{T}_1$ depends entirely on the rotation $\Delta\mathbf{R}_0$

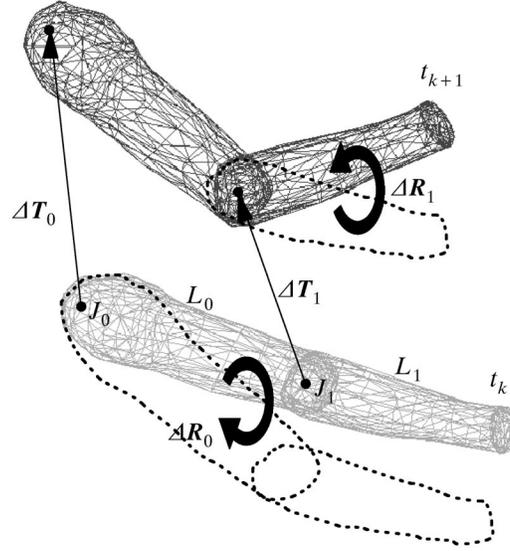


Figure 1: Example of the motion of the human arm model.

and translation vector $\Delta\mathbf{T}_0$ of the root link L_0 and it is computed as follows:

$$\Delta\mathbf{T}_1 = \Delta\mathbf{R}_0 \cdot (\mathbf{J}_1 - \mathbf{J}_0) + \mathbf{J}_0 + \Delta\mathbf{T}_0 - \mathbf{J}_1 . \quad (2)$$

According to Eq. 2 the motion of the link L_0 enforces a translation on the link L_1 . Since the translation vector can be calculated applying equation Eq. 2, only the rotation angles of the link L_1 need to be estimated. In following, an algorithm for estimating the translation vector, and the rotation angles of the root link L_0 , and the rotation angles of the link L_1 is described.

For estimating the motion parameters of the root link L_0 a set of N observation points $\mathbf{A}^{(n)}, 0 \dots n \dots N-1, N > 6$, are evaluated. An observation point lies on the surface of the mesh of the link at position $\mathbf{A}^{(n)} = (A_x^{(n)}, A_y^{(n)}, A_z^{(n)})^\top$ and carries the intensity value $I^{(n)}$ at this position. The position of the projection of the observation point $\mathbf{A}^{(n)}$ into the image plane is represented by $\mathbf{a}^{(n)} = (a_x^{(n)}, a_y^{(n)})^\top$. Let $\mathbf{g}^{(n)} = (g_x^{(n)}, g_y^{(n)})^\top$ be the observable linear intensity gradient at image position $\mathbf{a}^{(n)}$. 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 ($|\mathbf{g}^{(n)}| > \delta_1$) 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 $\mathbf{a}^{(n)}$ is approximated as follows:

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

where $I_k(\mathbf{a}^{(n)})$ and $I_{k+1}(\mathbf{a}^{(n)})$ represent the intensity value of the images I_k and I_{k+1} at the position $\mathbf{a}^{(n)}$, respectively. Since in general $\mathbf{a}^{(n)}$ lies outside of the image raster, the intensity value $I_{k+1}(\mathbf{a}^{(n)})$ is computed by bilinear interpolation of the intensity values of



the nearest four pixels in the intensity image I_{k+1} . The frame to frame intensity difference at N observation points is represented as follows:

$$\mathbf{FD}_0 = (fd(\mathbf{a}^{(N-1)}), fd(\mathbf{a}^{(N-2)}), \dots, fd(\mathbf{a}^{(0)}))^\top.$$

The motion 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 are estimated by maximizing the conditional probability $p(\mathbf{FD}_0|\mathbf{B}_0)$ of the frame to frame intensity differences \mathbf{FD}_0 at N observation points:

$$p(\mathbf{FD}_0|\widehat{\mathbf{B}}_0) \geq p(\mathbf{FD}_0|\mathbf{B}_0) \quad \forall \mathbf{B}_0, \quad (3)$$

where $\widehat{\mathbf{B}}_0 = (\widehat{\Delta T_x^{(0)}}, \widehat{\Delta T_y^{(0)}}, \widehat{\Delta T_z^{(0)}}, \widehat{\Delta \omega_x^{(0)}}, \widehat{\Delta \omega_y^{(0)}}, \widehat{\Delta \omega_z^{(0)}})^\top$ are the estimated motion parameters. To simplify the maximization Eq. 3 can be written as follows [14, 15]:

$$\frac{\partial \ln p(\mathbf{FD}_0|\mathbf{B}_0)}{\partial \mathbf{B}_0} \Big|_{\mathbf{B}_0=\widehat{\mathbf{B}}_0} = \frac{\partial ((\mathbf{FD}_0 - \mathbf{O}_0 \cdot \mathbf{B}_0)^\top \mathbf{U}_0^{-1} (\mathbf{FD}_0 - \mathbf{O}_0 \cdot \mathbf{B}_0))}{\partial \mathbf{B}_0} \Big|_{\mathbf{B}_0=\widehat{\mathbf{B}}_0} = 0.$$

Thus, the Maximum-Likelihood motion estimates $\widehat{\mathbf{B}}_0$ are given by:

$$\widehat{\mathbf{B}}_0 = (\mathbf{O}_0^\top \mathbf{U}_0^{-1} \mathbf{O}_0)^{-1} \mathbf{O}_0^\top \mathbf{U}_0^{-1} \mathbf{FD}_0, \quad (4)$$

where

$$\mathbf{O}_0 = (\mathbf{o}^{(N-1)}, \mathbf{o}^{(N-2)}, \dots, \mathbf{o}^{(0)})^\top,$$

$$\mathbf{o}^{(n)} = \begin{bmatrix} \frac{f \cdot g_x^{(n)}}{A_z^{(n)}} \\ \frac{f \cdot g_y^{(n)}}{A_z^{(n)}} \\ f \cdot (A_x^{(n)} g_x^{(n)} + A_y^{(n)} g_y^{(n)}) \\ \frac{f \cdot [A_x^{(n)} g_x^{(n)} (A_z^{(n)} - J_y^{(0)}) + A_y^{(n)} g_y^{(n)} (A_z^{(n)} - J_x^{(0)}) + A_z^{(n)} g_z^{(n)} (A_x^{(n)} - J_z^{(0)})]}{A_z^{(n)} A_z^{(n)}} \\ \frac{f \cdot [A_y^{(n)} g_y^{(n)} (A_x^{(n)} - J_x^{(0)}) + A_x^{(n)} g_x^{(n)} (A_x^{(n)} - J_x^{(0)}) + A_z^{(n)} g_z^{(n)} (A_z^{(n)} - J_z^{(0)})]}{A_z^{(n)} A_z^{(n)}} \\ \frac{f \cdot [g_x^{(n)} (A_y^{(n)} - J_y^{(0)}) - g_y^{(n)} (A_x^{(n)} - J_x^{(0)})]}{A_z^{(n)} A_z^{(n)}} \end{bmatrix}^\top,$$

$$\mathbf{U}_0 = \begin{bmatrix} \sigma_{\Delta I}^2(N-1) & 0 & 0 & \dots & 0 \\ 0 & \sigma_{\Delta I}^2(N-2) & 0 & \dots & 0 \\ 0 & 0 & \sigma_{\Delta I}^2(N-3) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \sigma_{\Delta I}^2(0) \end{bmatrix},$$

$$\sigma_{\Delta I}^2(n) = \frac{\sigma_z^2}{(A_z^{(n)})^2} \left(\left(g_x^{(n)} a_x^{(n)} + g_y^{(n)} a_y^{(n)} \right)^2 + \frac{f^2}{\sigma_z^2} \left(g_x^{(n)2} \sigma_x^2 + g_y^{(n)2} \sigma_y^2 \right) \right),$$

and σ_x^2 , σ_y^2 , and σ_z^2 are the variances of the position error at the observation points. The position error is assumed to be the result of the shape and motion estimation errors incurred by the motion analysis of previous frames. In order to improve the reliability



and accuracy of the estimates the Maximum-Likelihood motion estimation algorithm is applied iteratively.

The estimation of the rotation angles of the link L_1 is carried out in two steps. In the first step the translation vector is computed using Eq. 2. Then, the link L_1 is motion compensated using the computed translation vector. Let \mathbf{A} and \mathbf{A}'' be the position of an arbitrary point on the surface of the link L_1 before the motion (i.e., at time t_k), and after the translation enforced by the link L_0 , respectively. The position \mathbf{A}'' is computed as follows:

$$\mathbf{A}'' = \mathbf{A} + \Delta \mathbf{T}_1 . \quad (5)$$

The joint position also moves from \mathbf{J}_1 to \mathbf{J}_1'' according to the latter equation. Let \mathbf{A}' the position of an arbitrary point on the surface of a relative link L_1 after the motion, i.e., at time t_{k+1} . \mathbf{A}' is computed from $\Delta \mathbf{R}_0$ as follows:

$$\mathbf{A}' = \Delta \mathbf{R}_0 \cdot (\mathbf{A}'' - \mathbf{J}_1'') + \mathbf{J}_1'' . \quad (6)$$

According to the Eq. 6 the following identity is true: $\mathbf{J}_1' = \mathbf{J}_1''$.

In the second step, the rotation angles $\mathbf{B}_1 = (\Delta \omega_x^{(1)}, \Delta \omega_y^{(1)}, \Delta \omega_z^{(1)})^\top$ of the link L_1 are estimated also by applying an iterative Maximum-Likelihood motion estimation algorithm. For the estimation only observation points $(\mathbf{A}^{(m)}, 0 \dots m \dots M-1, M > 3)$ on the surface of the link L_1 are evaluated:

$$\widehat{\mathbf{B}}_1 = \left(\mathbf{O}_1^\top \mathbf{U}_1^{-1} \mathbf{O}_1 \right)^{-1} \mathbf{O}_1^\top \mathbf{U}_1^{-1} \mathbf{F} \mathbf{D}_1 , \quad (7)$$

where

$$\mathbf{F} \mathbf{D}_1 = (fd(\mathbf{a}^{(M-1)''}), fd(\mathbf{a}^{(M-2)''}), \dots, fd(\mathbf{a}^{(0)''}))^\top .$$

$$\mathbf{O}_1 = (\mathbf{o}^{(M-1)}, \mathbf{o}^{(M-2)}, \dots, \mathbf{o}^{(0)})^\top ,$$

$$\mathbf{o}^{(m)} = \begin{bmatrix} \frac{f \cdot [A_x^{(m)''} g_x^{(m)} (A_y^{(m)''} - J_y') + A_y^{(m)''} g_y^{(m)} (A_x^{(m)''} - J_x') + A_z^{(m)''} g_z^{(m)} (A_x^{(m)''} - J_x')] }{A_z^{(m)''} A_z^{(m)''}} \\ \frac{f \cdot [A_y^{(m)''} g_y^{(m)} (A_x^{(m)''} - J_x') + A_x^{(m)''} g_x^{(m)} (A_y^{(m)''} - J_y') + A_z^{(m)''} g_z^{(m)} (A_x^{(m)''} - J_x')] }{A_z^{(m)''} A_z^{(m)''}} \\ \frac{f \cdot [g_x^{(m)} (A_y^{(m)''} - J_y') - g_y^{(m)} (A_x^{(m)''} - J_x')] }{A_z^{(m)''}} \end{bmatrix}^\top ,$$

$$\mathbf{U}_1 = \begin{bmatrix} \sigma_{\Delta I^{(M-1)}}^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_{\Delta I^{(M-2)}}^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_{\Delta I^{(M-3)}}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \sigma_{\Delta I^{(0)}}^2 \end{bmatrix} ,$$

$$\sigma_{\Delta I^{(m)}}^2 = \frac{\sigma_z^2}{(A_z^{(m)''})^2} \left(\left(g_x^{(m)} a_x^{(m)''} + g_y^{(m)} a_y^{(m)''} \right)^2 + \frac{f^2}{\sigma_z^2} \left(g_x^{(m)2} \sigma_x^2 + g_y^{(m)2} \sigma_y^2 \right) \right) .$$

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 . 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 parallel to the image plane of the camera).



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. The ROBONAUT simulation matches the appearance and kinematics of the real ROBONAUT, and its state is controlled from other processes just like the real ROBONAUT. The control communication is done through a ROBONAUT API developed at NASA - JSC [3]. 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 are not valid. As its underlying communications package, the ROBONAUT API uses the commercial product Network Data Delivery Service (NDDS) by Real Time Innovations, Inc. (RTI). The graphics of the ROBONAUT simulation are created using the Enigma Core libraries available from NASA - JSC Interactive Graphics, Operations, and Analysis Laboratory (IGOAL)[9]. 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 their 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 enacts 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 six steps to each image I_{k+1} of the image sequence:

1. Estimate the motion parameters of the operator's right arm from time t_k to t_{k+1} . 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+1} as follows:

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

where \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 , respectively. The palm is assumed to be a rigid extension of the lower arm.

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



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

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} as follows:

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

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 and have performed a number of experiments on 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 running Windows 2000. The average processing time was 1.02 s per frame. The minimum and maximum processing times were 0.43 s and 8.65 s per frame, respectively. In all the experiments, the threshold δ_1 was set to 20. This value was experimentally determined. Due to the lack of space, we present the experimental results obtained from two real image sequences (HAZEL-A and HAZEL-C) only.

For the first 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. 2(a-d)). Figs. 2(e-h) depict the model at the estimated position and orientation overlaid at the image sequence. Fig. 3(a) depicts the ground truth (manually measured) and the positions (computed using the estimated motion parameters) of the right index finger for the subject depicted in the HAZEL-C sequence. Fig. 3(b) 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 cm². The mean and variance of the magnitude of the position error on the image plane are 1.099577 pixel and 0.586255 pixel², respectively. The minimum and the maximum values 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 are 0.043543 pixel and 3.142040 pixel, respectively.

For the second experiment, we tested the Maximum-Likelihood Motion Estimation Algorithm using the HAZEL-A sequence (200 frames) depicting a woman grasping and moving an object in front of a bookshelf. Figs. 4(a-d) depict the model at the estimated position and orientation overlaid at the image sequence. Figs. 4(e-h) and Figs. 4(i-l) depict the coronal and the sagittal view of the virtual ROBONAUT being animated with the estimated motion parameters of the image sequence HAZEL-A.

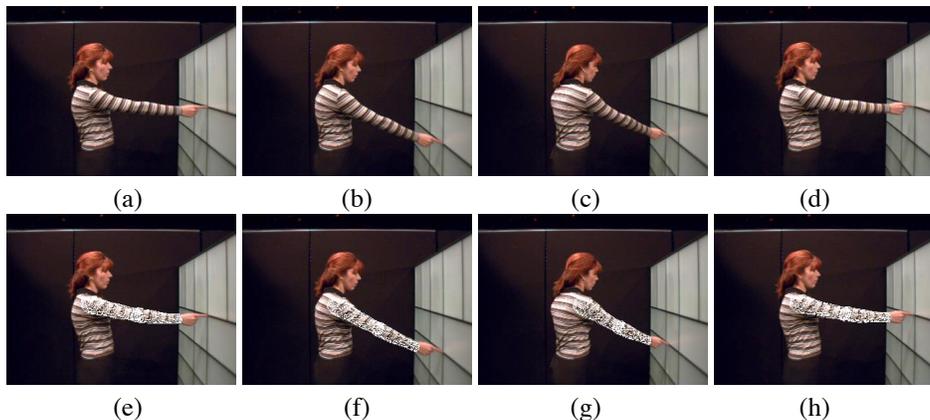


Figure 2: (a-d) Frames 1, 120, 180, and 354 from the sequence HAZEL-C, respectively. (e-h) Original frames with the model overlaid at the estimated position and orientation.

5 Conclusions

We have implemented the Maximum-Likelihood motion estimation algorithm of articulated objects and applied it for estimating the motion of a moving human arm. Then, we performed a number of experiments on real data to assess its accuracy, limitations and advantages. The experimental results revealed a position error for the right index finger of 0.57 ± 0.31 cm. 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.issu.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] R. O. Ambrose, H. Aldridge, R. S. Askew, R. R. Burrige, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark. Robonaut: Nasa's space humanoid. *IEEE Intelligent Systems*, 15(4):57–63, 2000.
- [3] B. Bluethmann. ROBONAUT API: Version 1.0. Manual, Dexterous Robotics Laboratory, NASA - Johnson Space Center, 2001.



- [4] 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.
- [5] 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.
- [6] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1), January 1999.
- [7] 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.
- [8] 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.
- [9] M. Goza. Enigma user manual. Manual, Interactive Graphics, Operations, and Analysis Laboratory (IGOAL), NASA - Johnson Space Center, 2001.
- [10] M. Goza. Robosim v2.2. Manual, Dexterous Robotics Laboratory, NASA - Johnson Space Center, 2001.
- [11] 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.
- [12] Ioannis A. Kakadiaris. Optical tracking for telepresence and teleoperation space applications. White paper, University of Houston, Houston, TX, 1999.
- [13] R. Koch. Dynamic 3D scene analysis through synthesis feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):556–568, June 1993.
- [14] G. Martinez. *Analyse-Synthese-Codierung basierend auf dem Modell bewegter dreidimensionaler, gegliederter Objecte*. PhD thesis, University of Hannover, Hannover, Germany, 1998.
- [15] 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, Seoul, Korea, April 25-27 2001.
- [16] 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.
- [17] 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.

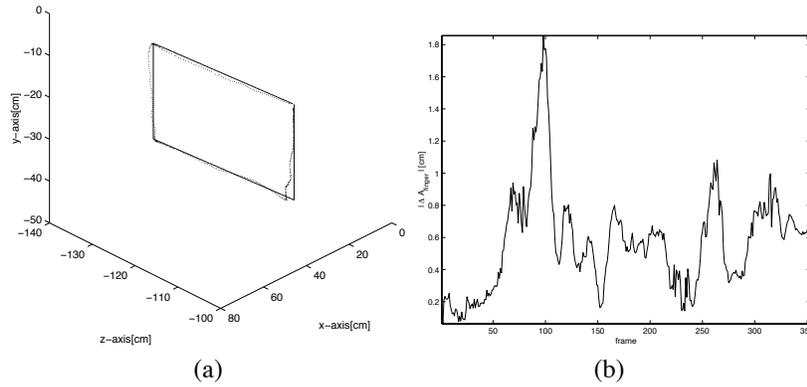


Figure 3: (a) Plot of the position of the right index finger of the subject depicted in the image sequence HAZEL-C. The solid line represents the ground truth position and the dotted one the computed position (using the motion estimates). (b) Plot of the magnitude of the position error of the index finger ($|\Delta \mathbf{A}_{finger}|$) for all the frames of the image sequence HAZEL-C.

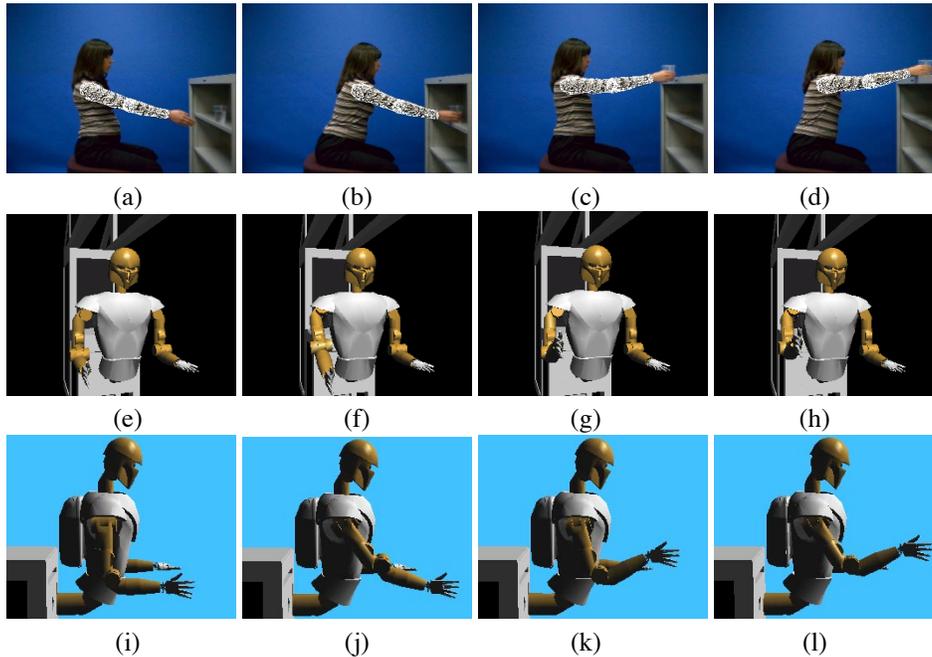


Figure 4: (a-d) Frames 1, 40, 116, and 160 from the sequence HAZEL-A with the model overlaid at the estimated position and orientation. (e-l) Commanding a ROBO-NAUT simulation developed at NASA - JSC using the estimated motion parameters of the HAZEL-A sequence. (e-h) Coronal and (i-l) sagittal view of the postures corresponding to the frames 1, 40, 116, and 160 of the sequence.