MULTIPLE VIEW TRACKING OF HUMANS MODELLED BY KINEMATIC CHAINS

Aravind Sundaresan, Rama Chellappa

Center for Automation Research
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742

Amit RoyChowdhury[*]

Department of Electrical Engineering
University of California, Riverside, CA 92521

ABSTRACT

We use a kinematic chain to model human body motion. We estimate the kinematic chain motion parameters using pixel displacements calculated from video sequences obtained from multiple calibrated cameras to perform tracking. We derive a linear relation between the 2D motion of pixels in terms of the 3D motion parameters of various body parts using a perspective projection model for the cameras, a rigid body motion model for the base body, and the kinematic chain model for the body parts. An error analysis of the estimator is provided, leading to an iterative algorithm for calculating the motion parameters from the pixel displacements. We provide experimental results to demonstrate the accuracy of our formulation. We also compare our iterative algorithm to the non-iterative algorithm and discuss its robustness in the presence of noise.

1. INTRODUCTION

Simultaneous 3D human shape estimation and motion tracking is a very challenging problem. There are methods that use silhouettes [1], edge information [2] or colour information [3] in order to track human body movement. These methods typically provide information only about the location of the body parts and not its exact posture. Given the complicated nature of human shape and motion, it can be very difficult to perform robust tracking of all body parts without using a motion and shape model. There are several models for human body shape modelling from stick figures and ellipsoids to more complicated models that are deformable [4]. Modelling the human body as rigid parts linked in a kinematic structure is a simple yet accurate model for tracking purposes. Optical flow can be exploited to provide dense information and obtain robust estimates of the motion parameters. Several papers [5, 6, 7, 8] have employed this set of motion parameters. However, [6] uses orthographic projection, while [7] uses a Bayesian formulation combined with a particle filtering approach to determine the motion parameters. Yamamoto, et al., [8] use a larger set of motion parameters to perform tracking using multiple views and perspective projection in an approximate formulation. We use kinematic chains to model human motion and arrive at a precise formulation for a tracker. We analyse the time-discretisation in a practical system and provide an algorithm for accurate estimation.

We illustrate our model for human structure and motion in Figure 1(a). We model the movement of human beings using kinematic chains with the root of the kinematic tree being the torso (base body). Each body part forearm, upper arm, torso, head, etc., is modelled as a rigid body, whose shape is known and is expressed in an object reference frame. At any given time we know the positions and orientations of the human body parts. We show that the instantaneous 3D velocity of each point in the base body coordinate system is a linear function of the vector $[V^{\mathsf{T}}, \dot{\boldsymbol{\theta}}^{\mathsf{T}}]^{\mathsf{T}}$, where V is the vector of base body motion parameters (rotation and translation), and $\boldsymbol{\theta}$ is the vector of angles describing the pose of the kinematic chain. In practice, however, we are able to measure only the 2D image pixel displacements at each pixel for different cameras. We show that, under perspective projection, the 2D velocities are still linear in the 3D motion parameters and we can accurately compute this linear relation. This enables us to accurately obtain the 3D motion parameters of the kinematic chain, given only the 2D pixel velocities obtained from multiple cameras. It is easier to obtain pixel displacement, $\Delta \boldsymbol{x}$, more accurately than pixel velocity $\boldsymbol{u} = \dot{\boldsymbol{x}}$.[1] Substituting pixel displacements for pixel velocities, we note that a bias is introduced in the estimation. We analyse the error introduced by this bias and propose an iterative algorithm by means of which we are able to decrease the bias.

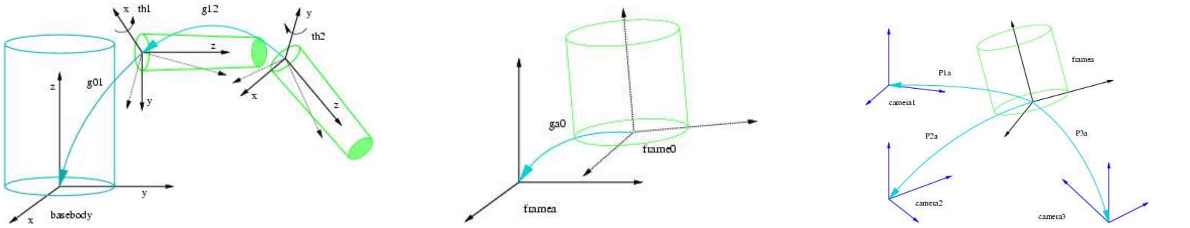
The experimental section presents evaluation results of our motion estimation performed on simulated data. We demonstrate the superior performance of the iterative algorithm as compared to the non-iterative algorithm in the absence of noise. We also present experimental results of the performance of the iterative algorithm in the presence of measurement noise.

2. OBTAINING THE 3D VELOCITY

We would like to estimate the motion parameters at time t_0. We define the base body reference frame, FRAME$_0$, that is attached to the base body (and moves with the base body) and a spatial reference frame, FRAME$_a$, that is static and coincides with FRAME$_0$ at time t_0 (see Figure 1(b)). We can, therefore, view the motion of the base body as the motion of FRAME$_0$. We consider a single kinematic chain of J body parts connected to the base body. (The equations that we derive can be trivially extended to multiple such chains.) Each body part is indexed by a number i and attached to a coordinate frame, FRAME$_i$. We parameterise the orientation between two connected components which possess a single degree of freedom in terms of the angle of rotation around the axis of the object coordinate frame, θ. We deal with joints that have multiple degrees of freedom by considering them as multiple joints with single degree of freedom. The transformation of a point from FRAME$_j$ coordinates to FRAME$_i$ coordinates is given by $g_{ij}(t)$.

[*] The third author performed the work while at the Center for Automation Research, University of Maryland, College Park.

[1] Since we have only a discrete set of images of the scene, estimation of arbitrary instantaneous velocities will be affected by the time-discretisation leading to an error in the estimates.



(a) Structure of Chain (b) Base body motion at time t (c) Projection onto Multiple Cameras

Fig. 1. Kinematic Chain Schematic

Consider a point on body part i in FRAME$_i$ coordinates, $\boldsymbol{q}_i^{(i)}$. The superscript denotes the index of the body part to which the point belongs and the subscript denotes the coordinate reference frame. We, therefore, have the relation $\boldsymbol{q}_a^{(i)}(t) = g_{a0}(t)\boldsymbol{q}_0^{(i)}(t)$ and

$$\dot{\boldsymbol{q}}_a^{(i)}(t) = \dot{g}_{a0}(t)\boldsymbol{q}_0^{(i)}(t) + g_{a0}(t)\dot{\boldsymbol{q}}_0^{(i)}(t) \triangleq \boldsymbol{v}_B(t) + \boldsymbol{v}_K(t). \quad (1)$$

The instantaneous velocity of this point, as given in (1), has two components: $\boldsymbol{v}_B(t) \triangleq \dot{g}_{a0}(t)\boldsymbol{q}_0^{(i)}(t)$, due to the motion of the base body itself, and $\boldsymbol{v}_K(t) \triangleq g_{a0}(t)\dot{\boldsymbol{q}}_0^{(i)}(t)$ due to the motion of the kinematic chain. We model these two motions differently in the following two subsections.

2.1. Velocity due to motion of base body

We consider the velocity component due to base body motion in this section. The motion of the base body is given by a rotation and a translation. From (1), dropping the point index i, we have the following, where $\boldsymbol{q}_a(t) = [X_a(t), Y_a(t), Z_a(t), 1]^{\mathsf{T}}$.

$$\begin{aligned} \boldsymbol{v}_B(t) &= \dot{g}_{a0}(t)\boldsymbol{q}_0(t) = \dot{g}_{a0}(t)g_{a0}^{-1}(t)\boldsymbol{q}_a(t) \\ &= \hat{V}_{a0}^s(t)\boldsymbol{q}_a(t) \end{aligned} \quad (2)$$

where $\hat{V}_{a0}^s(t) \triangleq \dot{g}_{a0}(t)g_{a0}^{-1}(t)$. We note that $g_{a0}(t)$ represents a rigid transformation and is given by $g_{a0}(t) = \begin{bmatrix} \mathsf{R}_{a0}(t) & \boldsymbol{p}_{a0}(t) \\ 0 & 1 \end{bmatrix}$. For simplicity, we drop the dependence on time. $\hat{V}_{a0}^s$ is called the spatial velocity. It is described by the translational velocity, $\boldsymbol{v}_{a0}^s = [v_1, v_2, v_3]^{\mathsf{T}}$, and the rotational velocity, $\boldsymbol{\omega}_{a0}^s = [\omega_1, \omega_2, \omega_3]^{\mathsf{T}}$. We can show that $\hat{V}_{a0}^s = \dot{g}_{a0}{g_{a0}}^{-1}$ $= \begin{bmatrix} \dot{\mathsf{R}}_{a0}\mathsf{R}_{a0}^{\mathsf{T}} & -\dot{\mathsf{R}}_{a0}\mathsf{R}_{a0}^{\mathsf{T}}\boldsymbol{p}_{a0} + \dot{\boldsymbol{p}}_{a0} \\ 0 & 0 \end{bmatrix}$. We define $V_{a0}^s = \begin{bmatrix} \boldsymbol{v}_{a0}^s \\ \boldsymbol{\omega}_{a0}^s \end{bmatrix} =$ $\begin{bmatrix} -\dot{\mathsf{R}}_{a0}\mathsf{R}_{a0}^{\mathsf{T}}\boldsymbol{p}_{a0} + \dot{\boldsymbol{p}}_{a0} \\ (\dot{\mathsf{R}}_{a0}\mathsf{R}_{a0}^{\mathsf{T}})^{\vee} \end{bmatrix}$, where $\hat{\boldsymbol{\omega}} \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$ and $(\hat{\boldsymbol{\omega}})^{\vee} \triangleq \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \boldsymbol{\omega}$. We can then reformulate $\hat{V}_{a0}^s\boldsymbol{q}_a$ as

$$\begin{aligned} \hat{V}_{a0}^s\boldsymbol{q}_a &= \begin{bmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_a \\ Y_a \\ Z_a \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & Z_a & -Y_a \\ 0 & 1 & 0 & -Z_a & 0 & X_a \\ 0 & 0 & 1 & Y_a & -X_a & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} V_{a0}^s \\ &= \mathsf{W}(\boldsymbol{q}_a)V_{a0}^s. \end{aligned} \quad (3)$$

Thus we see that the vector V_{a0}^s describes the motion of the base body. The instantaneous velocity of a point $\boldsymbol{q}_a$ due to base body motion is given by $\boldsymbol{v}_B = \hat{V}_{a0}^s\boldsymbol{q}_a = \mathsf{W}(\boldsymbol{q}_a)V_{a0}^s$.

2.2. Velocity due to motion of kinematic chain

Consider a point on body part i, $\boldsymbol{q}_i^{(i)}$ in FRAME$_i$ coordinates. The transformation of a point from FRAME$_i$ to FRAME$_0$ is represented by $g_{0i}(t)$. Then the relation between $\boldsymbol{q}_0^{(i)}(t)$ and $\boldsymbol{q}_i^{(i)}(t)$ is given by $\boldsymbol{q}_0^{(i)}(t) = g_{0i}(t)\boldsymbol{q}_i^{(i)}(t)$. Since the point is attached to FRAME$_i$, $\dot{\boldsymbol{q}}_i^{(i)}(t) = 0$. Then the instantaneous velocity of the point $\boldsymbol{q}_0^{(i)}$ can be obtained as

$$\dot{\boldsymbol{q}}_0^{(i)}(t) = \dot{g}_{0i}(t)\boldsymbol{q}_i^{(i)}(t) = \hat{V}_{0i}^s\boldsymbol{q}_0^{(i)}(t), \quad (4)$$

where $\hat{V}_{0i}^s$ is the spatial velocity and is equal to $\dot{g}_{0i}(t)g_{0i}^{-1}(t)$.

In the case of kinematic chains [9] $g_{(i-1)i}(t)$ has the form, $g_{(i-1)i}(t) = \bar{g}_{(i-1)i}e^{\hat{\xi}_i\theta_i(t)}$, where $\bar{g}_{(i-1)i} = g_{(i-1)i}(0)$ and $\hat{\xi}_i$ is a constant matrix that depends on which axis body part i rotates around. We therefore have $\dot{g}_{(i-1)i}(t) = \bar{g}_{(i-1)i}\hat{\xi}_i\dot{\theta}_i(t)e^{\hat{\xi}_i\theta_i(t)}$ and

$$\hat{V}_{(i-1)i}^s(t) = \dot{g}_{(i-1)i}(t)g_{(i-1)i}^{-1}(t) = \bar{g}_{(i-1)i}(t)\hat{\xi}_i\dot{\theta}_i(t)\bar{g}_{(i-1)i}^{-1}(t).$$

We drop the dependence on t in the following equations. Consider a kinematic chain with K links, where the motion of the k^{th} link is represented by θ_k. The reference frame (indexed by subscript 0) is attached to the base body, and FRAME$_k$, is attached to the k^{th} body part, for $k \in \{1, 2, \ldots, K\}$. Note that if a joint has multiple degrees of freedom, then the number of body parts (J) is less than the number of frames (K). We have $g_{0i} = g_{01}g_{12}\ldots g_{(i-1)i}$ and the spatial velocity is expanded as

$$\begin{aligned} \hat{V}_{0i}^s &= \dot{g}_{0i}g_{0i}^{-1} = \sum_{j=0}^{i-1} g_{0j}\left(\dot{g}_{j(j+1)}g_{j(j+1)}^{-1}\right)g_{0j}^{-1} \\ &= \sum_{j=0}^{i-1} g_{0j}\hat{V}_{j(j+1)}^s g_{0j}^{-1}. \end{aligned}$$

We note that $\boldsymbol{q}_0 = [X_0, Y_0, Z_0, 1]^{\mathsf{T}}$. We express the instantaneous velocity in terms of the kinematic chain motion parameters as

$$\begin{aligned} g_{a0}\dot{\boldsymbol{q}}_0 &= g_{a0}\hat{V}_{0i}^s g_{a0}^{-1}\boldsymbol{q}_a \\ &= g_{a0}\hat{V}_{01}^s g_{a0}^{-1}\boldsymbol{q}_a + g_{a0}\sum_{i=1}^{K-1} g_{0i}\hat{V}_{i(i+1)}^s g_{0i}^{-1}g_{a0}^{-1}\boldsymbol{q}_a \\ &= \begin{bmatrix} \hat{\xi}_1' & \hat{\xi}_2' & \cdots & \hat{\xi}_i' & 0 & \cdots & 0 \end{bmatrix}\dot{\boldsymbol{\theta}} \\ &= \mathsf{E}(\boldsymbol{q}_a, \boldsymbol{\theta})\dot{\boldsymbol{\theta}}, \end{aligned} \quad (5)$$

where $\mathsf{E}(\boldsymbol{q}_a, \boldsymbol{\theta}) \triangleq \left[\begin{array}{ccccccc} \hat{\xi}'_1 & \hat{\xi}'_2 & \cdots & \hat{\xi}'_i & 0 & \cdots & 0 \end{array}\right]$, and $\dot{\boldsymbol{\theta}} = \left[\begin{array}{cccc} \dot{\theta}_1 & \dot{\theta}_2 & \cdots & \dot{\theta}_K \end{array}\right]^\mathsf{T}$ and

$$\hat{\xi}'_j = \begin{cases} g_{a0} g_{0(j-1)} \bar{g}_{(j-1)j} \hat{\xi}_j \bar{g}^{-1}_{(j-1)j} g^{-1}_{0(j-1)} g^{-1}_{a0} \boldsymbol{q}_a & \text{if } j \leq i \\ 0 & \text{if } j > i. \end{cases}$$

2.3. Combined 3D velocity

We see that the complete 3D motion parameters are given by vectors V_{a0} and $\dot{\boldsymbol{\theta}}$. Combining equations (1), (2), and (5), we get

$$\dot{\boldsymbol{q}}_a(t) \quad = \quad \mathsf{M}(\boldsymbol{q}_a, \boldsymbol{\theta})\boldsymbol{\varphi}, \tag{6}$$

where $\mathsf{M}(\boldsymbol{q}_a, \boldsymbol{\theta}) \triangleq \left[\begin{array}{cc} \mathsf{W}(\boldsymbol{q}_a) & \mathsf{E}(\boldsymbol{q}_a, \boldsymbol{\theta}) \end{array}\right]$ and $\boldsymbol{\varphi} = \left[\begin{array}{c} V_{a0} \\ \dot{\boldsymbol{\theta}} \end{array}\right]$.

3. MULTIPLE CAMERA EQUATIONS

In section 2 we showed that the instantaneous 3D velocity at a point is a linear function of the motion parameters. In a practical setup we can only measure the motion of a point in the image coordinates of the camera. In this section, we show that the instantaneous image velocity of a pixel, under perspective projection, is also linear in the motion parameters. Assume there are C cameras, numbered from 1 through C. We use the superscript c to denote the camera index. Let $\mathsf{P}^{ca} = [\boldsymbol{p}^{ca}_1, \boldsymbol{p}^{ca}_2, \boldsymbol{p}^{ca}_3]^\mathsf{T}$ map a point in FRAME$_a$, $\boldsymbol{q}_a = [X_a, Y_a, Z_a, 1]^\mathsf{T}$ to the homogeneous image coordinates, $\tilde{\boldsymbol{q}}_c = [\tilde{x}_c, \tilde{y}_c, \tilde{z}_c]^\mathsf{T}$, of the c^{th} camera(see Figure 1(c)). We then have $\tilde{\boldsymbol{q}}_c = \mathsf{P}^{ca}\boldsymbol{q}_a$. The inhomogeneous image coordinates are given by $\left[\begin{array}{c} x_c \\ y_c \end{array}\right] = \frac{1}{\boldsymbol{p}^{ca\mathsf{T}}_3 \boldsymbol{q}_a} \left[\begin{array}{c} \boldsymbol{p}^{ca\mathsf{T}}_1 \\ \boldsymbol{p}^{ca\mathsf{T}}_2 \end{array}\right] \boldsymbol{q}_a$. $\dot{\boldsymbol{x}}_c$, the image coordinate velocities can be derived as follows.

$$\begin{aligned} \left[\begin{array}{c} \dot{x}_c \\ \dot{y}_c \end{array}\right] &= \frac{1}{\boldsymbol{p}^{ca\mathsf{T}}_3 \boldsymbol{q}_a} \left[\begin{array}{c} \boldsymbol{p}^{ca\mathsf{T}}_1 \\ \boldsymbol{p}^{ca\mathsf{T}}_2 \end{array}\right] \dot{\boldsymbol{q}}_a - \frac{\boldsymbol{p}^{ca\mathsf{T}}_3 \dot{\boldsymbol{q}}_a}{(\boldsymbol{p}^{ca\mathsf{T}}_3 \boldsymbol{q}_a)^2} \left[\begin{array}{c} \boldsymbol{p}^{ca\mathsf{T}}_1 \\ \boldsymbol{p}^{ca\mathsf{T}}_2 \end{array}\right] \boldsymbol{q}_a \\ &= \frac{1}{\tilde{z}_c} \left[\begin{array}{c} \boldsymbol{p}^{ca\mathsf{T}}_1 - (\tilde{x}_c/\tilde{z}_c)\boldsymbol{p}^{ca\mathsf{T}}_3 \\ \boldsymbol{p}^{ca\mathsf{T}}_2 - (\tilde{y}_c/\tilde{z}_c)\boldsymbol{p}^{ca\mathsf{T}}_3 \end{array}\right] \dot{\boldsymbol{q}}_a \\ &\triangleq \mathsf{C}^c(\boldsymbol{q}_a, \mathsf{P}^{ca})\dot{\boldsymbol{q}}_a \end{aligned} \tag{7}$$

Combining (6) and (7), we get the following comprehensive equation.

$$\dot{\boldsymbol{x}}_c = \mathsf{C}^c(\boldsymbol{q}_a, \mathsf{P}^{ca})\mathsf{M}(\boldsymbol{q}_a, \boldsymbol{\theta})\boldsymbol{\varphi} \tag{8}$$

We see that the number of unknowns is fixed, irrespective of the number of points and the number of cameras, and is equal to $6+K$, where K is the number of degrees of freedom of the kinematic chain. If we can use $N(c)$ points from the c^{th} camera, then we have a total of $2\sum_{c=1}^{C} N(c)$ equations. Note that we need pixels on the k^{th} body part to estimate $\dot{\theta}_1, \cdots, \dot{\theta}_k$.

4. ESTIMATOR AND ERROR ANALYSIS

We have shown that the 2D velocity of a pixel is a linear function of our motion parameter vector. For the purpose of error analysis, we can simplify (dropping the camera subscript) the form of equation (8) to $\dot{\boldsymbol{x}} = \mathsf{A}(t)\boldsymbol{\varphi}$, where $\mathsf{A}(t) = \mathsf{C}^c(\boldsymbol{q}_a, \mathsf{P}^{ca})\mathsf{M}(\boldsymbol{q}_a, \boldsymbol{\theta})$. If there are a total of N points and K degrees of freedom in the kinematic chain, then $\dot{\boldsymbol{x}}$ is $2N \times 1$, A is $2N \times M$, and $\boldsymbol{\varphi}$ is $M \times 1$, where $M = K + 6$. Since we work with pixel displacements $(\Delta\boldsymbol{x})$ instead of velocities, we formulate our estimator accordingly and analyse the error introduced due to the approximation. We can expand the pixel position in terms of the Taylor series and thus obtain $\Delta\boldsymbol{x}$ as

$$\Delta\boldsymbol{x} \triangleq \boldsymbol{x}(t + \Delta t) - \boldsymbol{x}(t) = \dot{\boldsymbol{x}}(t)\Delta t + \mathcal{O}(\Delta t^2).$$

We drop the dependence on time in the following equations. There is also some measurement noise in the pixel displacement and we model the measured pixel displacement, $\Delta\tilde{\boldsymbol{x}}$, as $\Delta\tilde{\boldsymbol{x}} = \Delta\boldsymbol{x} + \boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is zero-mean measurement noise with variance Σ. We therefore have the following equations.

$$\Delta\tilde{\boldsymbol{x}} = \dot{\boldsymbol{x}}(t)\Delta t + \mathcal{O}(\Delta t^2) + \boldsymbol{\eta} = \mathsf{A}\boldsymbol{\varphi}\Delta t + \mathcal{O}(\Delta t^2) + \boldsymbol{\eta} \tag{9}$$

If we ignore second and higher order terms, then we have

$$\Delta\tilde{\boldsymbol{x}} = \mathsf{A}\boldsymbol{\varphi}\Delta t + \boldsymbol{\eta}. \tag{10}$$

Assuming $\Delta t = 1$, an unbiased estimator for $\boldsymbol{\varphi}$ in (10), $\hat{\boldsymbol{\varphi}} = (\mathsf{A}^\mathsf{T}\mathsf{A})^{-1}\mathsf{A}^\mathsf{T}\Delta\tilde{\boldsymbol{x}}$, is obtained by minimising the cost function $C(\hat{\boldsymbol{\varphi}}) = \frac{1}{2}||\Delta\tilde{\boldsymbol{x}} - \mathsf{A}^\mathsf{T}\hat{\boldsymbol{\varphi}}||$. The higher order terms can be interpreted as a bias that affects the estimator. We propose an iterative algorithm to eliminate this bias. In our iterative algorithm, we first estimate the parameters using the linear equation. We then compute the 2D motion (pixel displacements) for the estimated motion. The computation of 2D motion, given the motion parameters is exact. We then compute the difference between the observed pixel displacements and the computed pixel displacements (using our current estimated motion parameters) and try to estimate a new set of motion parameters using the error in the displacements. As we thus get closer to the true solution, the linear estimation becomes more accurate ($\mathcal{O}(\Delta t^2)$ tends to zero) and we ultimately arrive at a fairly accurate estimate of the true motion parameters. In the case of Gaussian noise ($\boldsymbol{\eta}$), we can show that $\hat{\boldsymbol{\varphi}} = (\mathsf{A}^\mathsf{T}\mathsf{A})^{-1}\mathsf{A}^\mathsf{T}\Delta\tilde{\boldsymbol{x}}$ is the Minimum Variance Unbiased Estimator (MVUE). Since we minimise the cost function $\frac{1}{2}||\Delta\tilde{\boldsymbol{x}} - \mathsf{A}^\mathsf{T}\hat{\boldsymbol{\varphi}}||$, we can obtain an expression for the error covariance $\mathrm{R}_{\hat{\boldsymbol{\varphi}}}$ as (page 34 of [10])

$$\mathrm{R}_{\hat{\boldsymbol{\varphi}}} = \mathrm{H}^{-1}\left(\sum_{i=1}^{2N} \mathsf{a}_i \mathsf{a}_i^\mathsf{T} r_i^{\Delta\tilde{\boldsymbol{x}}}\right)\mathrm{H}^{-\mathsf{T}},$$

where $\mathrm{H} = \sum_{i=1}^{2N} \mathsf{a}_i \mathsf{a}_i^\mathsf{T}$, $\mathrm{R}_{\Delta\tilde{\boldsymbol{x}}} = \mathrm{diag}(r_1^{\Delta\tilde{\boldsymbol{x}}}, r_2^{\Delta\tilde{\boldsymbol{x}}}, \cdots, r_{2N}^{\Delta\tilde{\boldsymbol{x}}})$ and a_i^T is the i^{th} row of A.

5. EXPERIMENTAL RESULTS

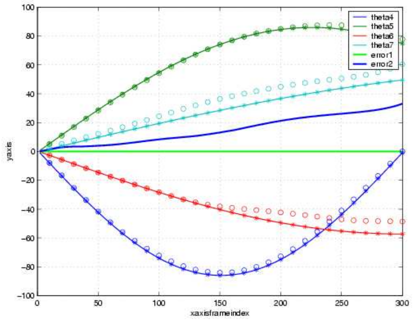
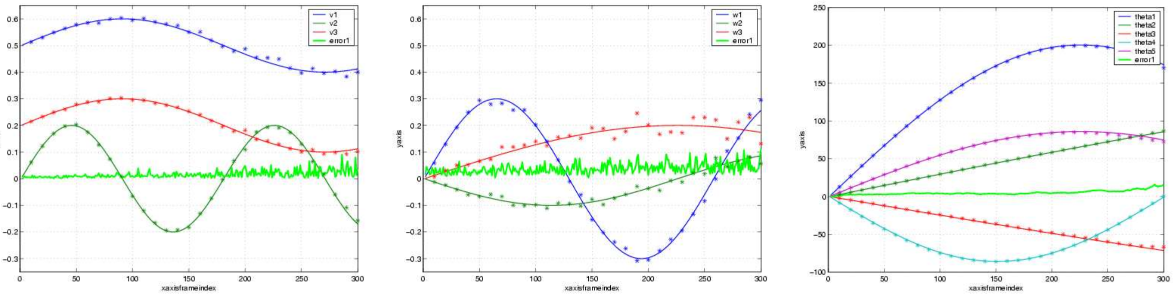


Fig. 2. Comparison of Iterative and Non-Iterative Algorithms.



(a) Base body velocity estimation (b) Base body angular velocity estimation (c) Kinematic chain angle estimation

Fig. 3. Estimation of Motion Parameters using iterative algorithm in the presence of measurement noise.

In our experiments we determine how our estimator for 3D motion parameters performs on simulated data. We have built a kinematic chain composed of a base body with three articulated objects attached in a kinematic chain as in Figure 1 (a). The base body moves with translational and rotational motion and the objects are constrained to rotate about their joints. There are 3 joints with 3, 2, and 2 degrees of freedom respectively. We project the 3D model onto $C = 3$ static cameras and record the pixel displacement for pixels on each of the four body parts in the C cameras. We can project the current pose onto the image to associate each pixel with the body part and thus handle occlusions. We first evaluate the performance of the tracker in the iterative and non-iterative modes on data that does not contain any noise. In our formulation for the non-iterative estimator, we do not account for the substitution of $\Delta \boldsymbol{x}_c$ for $\boldsymbol{u}_c$. This is an oft-used approximation and is valid for small motions. However, in a tracking problem errors accumulate and this can lead to the object ultimately losing track of the object. Indeed, in our experiments, we find that in the non-iterative tracker, the errors accumulate quickly despite the absence of any noise. This is because the non-iterative formulation is not exact. Figure 2 compares the output of the non-iterative and the iterative algorithms. The "$*$" markers are the outputs of the the iterative algorithm and "$\circ$" markers are the outputs of the the non-iterative algorithm. Our experiments show that our iterative algorithm results in a near perfect tracker while the non-iterative algorithm deteriorates with time. Error_1 is the tracking error of the iterative algorithm and Error_2 is the error of the non-iterative algorithm in this and subsequent figures. Only some of the motion parameters are plotted in some of the figures for clarity. We also note that the iterative estimator converges faster if we estimate the parameters sequentially starting from the root of the kinematic chain.

In Figure 3, we present the performance of the tracker in the iterative mode on data that contains measurement noise. We add uncorrelated zero mean Gaussian noise with $\sigma = 0.0006$ (note that in the system the average pixel displacement is 0.0034) to the pixel displacement values. We see that the tracker follows the actual parameter value for most parameters. We see that the base body motion parameter estimates are the poorest. This is to be expected because the ratio of the number of parameters estimated to the number of equations available is the least for the base body. One would expect that the estimates will be more robust when there are are more pixels on the target. We find that the estimates of the kinematic chain motion parameters are very robust for the same reason.

6. CONCLUSION

We have outlined an algorithm for estimating 3D motion parameters of a kinematic chain model from 2D image pixel displacements. We use a kinematic chain model for human motion and use the perspective projection model and multiple cameras in our formulation. We estimate the 3D motion parameters from pixel displacements. We show that the estimation of 3D motion parameters is biased and propose an iterative algorithm that results in a very accurate tracker when applied on simulated data. We also report results on simulated data that contains measurement noise and find that the iterative algorithm to be robust and successfully tracks the motion parameters. The compact form of the 3D motion parameter vector, and the linear formulation make it easy to extend it into a stochastic formulation (Kalman filter) which will provide robustness when applied to real data and will also exploit the temporal correlation of the motion parameters.

7. REFERENCES

[1] Q. Delamarre and O. Faugeras, "3D articulated models and multi-view tracking with silhouettes," in *ICCV*, 1999, pp. 716–721.

[2] D.M. Gavrila and L.S. Davis, "3-D model-based tracking of humans in action: A multi-view approach," in *Computer Vision and Pattern Recognition*, 1996, pp. 73–80.

[3] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE PAMI*, vol. 19, pp. 780–785, July 1997.

[4] I. Kakadiaris and D. Metaxas, "Model-based estimation of 3D human motion," *IEEE PAMI*, pp. 1453–1459, 2000.

[5] M. Yamamoto and K. Koshikawa, "Human motion analysis based on a robot arm model," in *CVPR*, 1991, pp. 664–665.

[6] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *CVPR*, 1998, pp. 8–15.

[7] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet, "Stochastic tracking of 3D human figures using 2D image motion," in *ECCV*, 2000, pp. 702–718.

[8] M. Yamamoto, A. Sato, S. Kawada, T. Kondo, and Y. Osaki, "Incremental tracking of human actions from multiple views," in *CVPR*, 1998, pp. 2–7.

[9] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulations*, CRC Press, 1994.

[10] A. RoyChowdhury and R. Chellappa, "Stochastic approximation and rate distortion analysis for robust structure and motion estimation," *IJCV*, pp. 27–53, October 2003.