to be published in IEEE CVPR'00, June 2000

Articulated-pose estimation using brightness- and depth-constancy constraints

Michele M. Covell
covell@interval.com

Ali Rahimi
ali@mit.edu

Michael Harville
harville@interval.com

Trevor J. Darrell
trevor@ai.mit.edu

Interval Research Corporation
1801 Page Mill Road, Bldg. C, Palo Alto, CA 94304, USA

Abstract

This paper explores several approaches for articulated-pose estimation, assuming that video-rate depth information is available, from either stereo cameras or other sensors. We use these depth measurements in the traditional linear brightness constraint equation, as well as in a depth constraint equation. To capture the joint constraints, we combine the brightness and depth constraints with twist mathematics. We address several important issues in the formation of the constraint equations, including updating the body rotation matrix without using a first-order matrix approximation and removing the coupling between the rotation and translation updates. The resulting constraint equations are linear on a modified parameter set. After solving these linear constraints, there is a single closed-form non-linear transformation to return the updates to the original pose parameters. We show results for tracking body pose in oblique views of synthetic walking sequences and in moving-camera views of synthetic jumping-jack sequences. We also show results for tracking body pose in side views of a real walking sequence.

1 Introduction

In this paper, we extend the head-pose tracking of Harville et al. [1] to articulated-pose tracking. We assume that we have video-rate depth images, from either stereo cameras or from other sensors. The depth images allow us to use depth-constancy constraint equations (ZCCE) that are similar to the classic brightness-constancy constraint equations (BCCE). The depth images also give us *linear* constraints, even when we use a perspective-camera model.

In Section 3, we review these constraint equations and use twist mathematics [2] to capture the motion constraints imposed by the articulated joints. Our basic twist derivations are similar in spirit to the derivations of Bregler et al. [3]. The primary differences trace back to the approximations made within the derivations: Bregler approximates perspective constraints using scaled-orthographic constraints and he approximates the body-rotation matrix using an extra first-order Taylor-series expansion. We avoid this first-order approximation by solving our constraints on a transformed parameter set and by remapping our results into the original parameter set using a closed-form non-linear function (Section 3.5).

Throughout Section 3, we assume that we know which limb each pixel corresponds to. To get this information, we must create limb-assignment maps. We describe the process that we use to do this in Section 4.

In Section 5, we re-derive BCCE and ZCCE using shifted centers of expansion in their Taylor-series approximations. This extension allows us to use these constraints on large motions without iteration. We also explicitly modify the formulation of our constraint matrices to decouple the body rotation and the body translation updates (Section 6). Finally, in Section 7, we present quantitative analyses of our results on synthetic sequences and qualitative results on real sequences.

2 Previous Work in Articulated-Pose Estimation

There have been many proposed techniques for tracking articulated-body motion [4][8][9][10][11][12]. Some approaches use constraints from widely separated views to disambiguate the partially occluded motions without computing depth values [6][10]. The most robust of these tend to fit the observed (dense) motion data to a parametric model before assigning specific pointwise correspondences between the images [3][5][6][7]. Typically, this approach results in non-linear constraint equations which must be solved using iterative gradient descent or relaxation methods [5][7].

Bregler et al. [3] and Yamamoto et al [6] provide notable exceptions to this general trend: both wind up with systems of linear constraint equations, created by combining articulated-body models with dense optical flow.

Yamamoto maintains the constraints between limbs by sequentially estimating the motion of each parent limb, adjusting the hypothesized position of the child limb, then estimating the further motion of the child limb. This is conceptually simpler than the approach taken by Bregler, but results in fewer constraints on the motion of the parent limbs. In contrast, Bregler takes full advantage of the information provided by child limbs to further constrain the estimated motions of the parents.

Both Yamamoto and Bregler use a first-order Taylor series approximation to the camera/body rotation matrix, to reduce the number of parameters used to represent this matrix. Furthermore, both use an articulated model to *generate* depth values that are needed to linearize the mapping from 3D body motions to observed 2D camera-plane motions.

This paper differs from this prior work on both of these counts. We do not use a first-order Taylor series approximation to the camera/body rotation matrix. Nor do we use an articulated figure model to generate depth values: instead, we assume that real-time depth images are available.

3 Articulated-Body Constraint Equations

In this section, we derive the basic articulated-body constraint equations.

3.1 Brightness- and depth-constancy constraints

We start from the well known BCCE:

$$\begin{aligned} B(x, y, t+1) &= B(x-v_x, y-v_y, t) \\ &\approx B(x, y, t) - v_x B_x(x, y, t) - v_y B_y(x, y, t) \end{aligned}$$

where v_x and v_y are the motions in X and Y, after projection onto the image plane.

Assuming that we have real-time depth images available to us, we can use a similar constraint equation on depth [1]. Combining BCCE and ZCCE, we have:

$$-\begin{bmatrix} B_t(x, y, t) \\ Z_t(x, y, t) \end{bmatrix} \approx \begin{bmatrix} B_x(x, y, t) & B_y(x, y, t) & 0 \\ Z_x(x, y, t) & Z_y(x, y, t) & -1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_Z \end{bmatrix} \qquad (1)$$

where v_Z is the motion in world coordinate Z.

Equation (1) gives us $2N$ equations in terms of $3N$ unknowns (v_x, v_y, and v_Z for each point) where N is the number of visible points on the articulated figure.

3.2 Perspective-camera constraints

In order to translate image-plane velocities into world-coordinate velocities, we need a camera model. Using a perspective camera model and assuming that the origin of the world coordinate system is at the camera and the z-axis is along the viewing axis of the camera, so that $x = f_x X/Z$, and $x = f_y Y/Z$, we get

$$\begin{bmatrix} v_x \\ v_y \\ v_Z \end{bmatrix} = \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{x}{Z} \\ 0 & \frac{f_y}{Z} & -\frac{y}{Z} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix} \qquad (2)$$

where v_X and v_Y are motions in world coordinates.

Equation (2) changes the $3N$ unknown parameters in our $2N$ constraint equations to new set of unknowns (v_X, v_Y, and v_Z for each point).

3.3 Rigid-limb constraints

We need to translate the velocities of the visible points on the figure into rotations and translations of the figure and its limbs relative to the world coordinates. To do this, we use twist mathematics [2].

A twist, $\xi = \begin{bmatrix} v^T & \omega^T \end{bmatrix}^T$, is a 6-element vector with the first 3 elements, v, (indirectly) representing the translation and the last three elements ω representing the axis (and sometimes the amount) of rotation. As a matter of convention, if the twist is used with an explicit scaling term θ, then $|\omega| = 1$; otherwise, the magnitude of ω is set according to the amount of rotation. The twist can be used to form a 4x4 matrix, through the operation of the "hat operator": $\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}$ where $\hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$. When exponentiated, this 4x4 matrix gives the rotation/translation matrix $e^{\hat{\xi}\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & p \\ 0 & 1 \end{bmatrix}$ where $e^{\hat{\omega}\theta}$ is the rotation and $p = ((I - e^{\hat{\omega}\theta})\hat{\omega} + \omega\omega^T\theta)v$ is the translation.

Using twists, the world coordinates (q_s) of any point on the body can be expressed as a function of time, of the point's limb number (k), of the pose parameters (ξ_0 and θ), and of the point's limb-centric coordinates (q_b):

$$q_s(t) = g_{sb}(\xi_0(t), \theta(t) | k, \xi_1, \ldots, \xi_{K-1}) q_b$$

where K is the number of articulated limbs in the figure. The mapping from limb-centric coordinates to world coordinates is done by translation/rotation as dictated by a "reference configuration" for the k^{th} limb, $g_{sk}(0)$; by the translations/rotations $e^{\hat{\xi}_1\theta_1} \cdots e^{\hat{\xi}_{k-1}\theta_{k-1}}$ introduced by the joints along the articulated chain up to the k^{th} limb; and by the translations/rotations $e^{\hat{\xi}_0}$ from the camera to the figure's torso.

Each limb's reference configuration gives the translation and rotation from that limb's coordinate system to the world coordinate system, when the body is positioned at $\xi_0 = 0$ and when all of the joint angles are zero. The extra degrees of freedom given by the reference configuration simplifies the task of describing the geometry of the articulated joint locations. In Section 5, we also use these extra degrees of freedom to decouple our body rotation and body translation estimates. Given a specific pose, the transformation from the limb's coordinate frame to the world coordinate frame is:

$$\begin{aligned} & g_{sb}(\xi_0(t), \theta(t) | k, \xi_1, \ldots, \xi_{K-1}) \\ &\qquad = e^{\hat{\xi}_0} e^{\hat{\xi}_1\theta_1} \cdots e^{\hat{\xi}_{k-1}\theta_{k-1}} g_{sk}(0) \end{aligned} \qquad (3)$$

For the remainder of this paper, for notational simplicity, we will refer to $g_{sb}(\xi_0(t), \theta(t) | k, \xi_1, \ldots, \xi_{K-1})$ simply as g_{sb}.

This description of the world coordinates of each body point in terms of the articulated-pose parameters relates the world velocities to the rotations and translations of the K coordinate frames that are tied to the K limbs of the figure. Recalling that $q_s(t) = g_{sb} q_b$, note that q_b is independent of time:

$$\begin{aligned} \begin{bmatrix} v_X & v_Y & v_Z & 0 \end{bmatrix}^T &= \frac{d}{dt} q_s(t) = \frac{\partial}{\partial t} g_{sb} q_b \\ &= \left(\frac{\partial}{\partial t} g_{sb}\right)(g_{sb}^{-1} q_s(t)) = \left(\frac{\partial}{\partial t} g_{sb} g_{sb}^{-1}\right) q_s(t) \\ &= \hat{V}_{sb}^{s} q_s(t) \end{aligned} \qquad (4)$$

The second line of the above identity is derived from the inverse of the identity $q_s(t) = g_{sb} q_b$. The third line is by definition: $\hat{V}^s_{sb} = \dot{g}_{sb}g^{-1}_{sb}$. $\hat{V}^s_{sb}$ is a 4x4 matrix describing the motion of the k^{th} limb's coordinate frame relative to the world coordinate frame, in terms of world coordinates. Using V^s_{sb}, a 6x1 vector, to describe this coordinate transformation makes use of the special structure of $\dot{g}_{sb}g^{-1}_{sb}$: namely, that the first three rows and columns of $\dot{g}_{sb}g^{-1}_{sb}$ are skew symmetric and the bottom row is all zeros [2]. $q_s(t) = \begin{bmatrix} q_X & q_Y & q_Z & 1 \end{bmatrix}^T$ is the (homogenous) world coordinates of the body point at time t. More generally, q_X, q_Y, and q_Z are the coordinates of the point within a coordinate system that is tied to the world coordinate system by some known translation (and rotation). We will use these extra degrees of freedom in Section 6 to improve the condition number of our constraint matrix. Rewriting Equation (4),

$$\begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & q_Z & -q_Y \\ 0 & 1 & 0 & -q_Z & 0 & q_X \\ 0 & 0 & 1 & q_Y & -q_X & 0 \end{bmatrix} V^s_{sb} \tag{5}$$

At this point, we have forced each limb to be internally rigid. This reduces the number of unknown parameters in our $2N$ constraint equations down to 6 parameters per limb (the 6 elements of V^s_{sb} for that limb).

3.4 Joint constraints

We can further constrain the problem by taking advantage of the interconnections between limbs. To do this, we describe V^s_{sb} for the k^{th} limb in terms of the articulated-pose parameters:

$$V^s_{sb} = V^s_{s0} + \sum_{i=1}^{k} J^s_{i-1,i}(\theta_i)\dot{\theta}_i \tag{6}$$

where V^s_{s0} is the velocity due to the motion of the body relative to the world coordinates and $J^s_{i-1,i}(\theta_i)\dot{\theta}_i$ is the velocity due to the motion of the i^{th} joint along the articulated chain to the k^{th} limb. Using the identity $\hat{V}^s_{sb} = \dot{g}_{sb}g^{-1}_{sb}$, we obtain

$$\hat{J}^s_{i-1,i}(\theta_i) = \left(\frac{\partial}{\partial\theta_i}g_{sb}\right)g^{-1}_{sb}$$

$$= e^{\hat{\xi}_0}e^{\hat{\xi}_1\theta_1}\cdots e^{\hat{\xi}_{i-1}\theta_{i-1}}\hat{\xi}_i e^{-\hat{\xi}_{i-1}\theta_{i-1}}\cdots e^{-\hat{\xi}_1\theta_1}e^{-\hat{\xi}_0}$$

To simplify this further, we introduce the adjoint of a rotation/translation matrix, $g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$. The adjoint is $\mathrm{Adj}(g) = \begin{bmatrix} R & \hat{p}R \\ 0 & R \end{bmatrix}$ and $g\hat{\xi}g^{-1} = (\mathrm{Adj}(g)\xi)^\wedge$ [2] where $(\ )^\wedge$ applies the hat operator to the vector contained within the parentheses. Using this identity,

$$J^s_{i-1,i}(\theta_i) = \mathrm{Adj}\left(e^{\hat{\xi}_0}e^{\hat{\xi}_1\theta_1}\ldots e^{\hat{\xi}_{i-1}\theta_{i-1}}\right)\xi_i \tag{7}$$

3.5 Body motion constraints and re-parameterization

The velocity vector of the figure relative to the world coordinates must allow for unconstrained rotations and translations. The easiest way to do this is to express these motions in terms of the 4x4 transformation matrix, instead of in terms of the twist coordinates. Let $e^{\hat{\xi}_0} = \begin{bmatrix} R_0 & p_0 \\ 0 & 1 \end{bmatrix}$. Then:

$$\hat{V}^s_{s0} = \left(\frac{d}{dt}e^{\hat{\xi}_0}\right)e^{-\hat{\xi}_0} = \begin{bmatrix} \dot{R}_0 R_0{}^T & -\dot{R}_0 R_0{}^T p_0 + \dot{p}_0 \\ 0 & 0 \end{bmatrix}$$

This constraint is linear in terms of the unknowns ($\dot{R}_0$ and $\dot{p}_0$) but it is not as tightly constrained as we would like: $\dot{R}_0$ has 9 unknowns, instead of the 3 unknowns that we know can be used to describe a rotation or its derivative. We correct this over-parameterization by noting that the first 3 rows/columns of $\hat{V}^s_{s0}$ must be skew symmetric. To capture this structure, we name the rotational component of the frame velocity: $\omega_{v_0} = V^s_{s0,4:6} = (\dot{R}_0 R_0{}^T)^\vee$ where $(\ )^\vee$ is the inverse of the hat operator on the skew-symmetric matrix contained within the parentheses.

Note that this is *not* a "small angle approximation" such as is often used for mapping a rotation matrix down onto its rotation axis [3]. The identity $\omega_{v_0} = (\dot{R}_0 R_0{}^T)^\vee$ is *exact*, due to the special structure embedded in the derivative of an orthonormal matrix, when that matrix is constrained to remain orthonormal. Under this orthonormality constraint, the derivative matrix times the transpose of the orthonormal matrix is a skew-symmetric matrix [2].

Substituting $\hat{\omega}_{v_0}$ for $\dot{R}_0 R_0{}^T$ and rearranging gives

$$V^s_{s0} = \begin{bmatrix} I & \hat{p}_0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \dot{p}_0 \\ \omega_{v_0} \end{bmatrix} \tag{8}$$

We now use our $2N$ linear constraint equations to solve for ω_{v_0}, $\dot{p}_0$, and $\dot{\theta}_1$ through $\dot{\theta}_{K-1}$ ($K+5$ unknowns) and then remap ω_{v_0} back into $\dot{R}_0$ according to

$$\dot{R}_0 = \hat{\omega}_{v_0} R_0{}^{-T} = \hat{\omega}_{v_0} R_0 \tag{9}$$

3.6 Discrete-time approximations to derivatives

Finally, we need discrete-time approximations to the time derivatives, $\dot{R}_0$, $\dot{p}_0$, and $\dot{\theta}_1$ through $\dot{\theta}_{K-1}$. We use a forward-difference approximation to the body-translation and joint-angle derivatives:

$$\begin{aligned} \dot{p}_0(t) &\rightarrow p_0(t+1) - p_0(t) \\ \dot{\theta}_i(t) &\rightarrow \theta_i(t+1) - \theta_i(t) \end{aligned} \tag{10}$$

We can not use a forward-difference approximation to the body-rotation derivative $\dot{R}_0$ since using this approximation destroys the orthonormal structure of the rotation matrix R_0. Instead, we must use a central-difference approximation:

$$\dot{R}_0\left(t + \frac{1}{2}\right) \rightarrow R_0(t+1) - R_0(t)$$

Combining this central difference approximation with Equation (9) and a linear-interpolation approximation for the half-sample delay, we get

$$R_0(t+1) - R_0(t) = \hat{\omega}_{v_0}\left(\frac{R_0(t) + R_0(t+1)}{2}\right)$$

so that

$$R_0(t+1) = \left(I - \frac{\hat{\omega}_{v_0}}{2}\right)^{-1}\left(I + \frac{\hat{\omega}_{v_0}}{2}\right)R_0(t) \qquad (11)$$

Combining equations (1), (2), (5), (6), (7), (8) and (10), we have, as before, $2N$ linear constraint equations in terms of $K+5$ unknowns. The difference is that the unknowns are now the updated parameters (ω_{v_0}, $p_0(t+1)$, $\theta_1(t+1)$ through $\theta_{K-1}(t+1)$). We solve these constraints using least squares. Once we have this solution, equation (11) provides the non-linear mapping from ω_{v_0} to $R_0(t+1)$.

4 Limb-assignment maps

Throughout our derivation of our constraint equations, we assumed that we knew which pixels were on which limb of the articulated figure and which pixels were not on the articulated figure. We must create such a limb-assignment map for each frame of our sequence. This section describes that process.

4.1 Creating the initial limb-assignment map

Creating the first limb-assignment map is easy since we are given the initial pose. We create the first lassignment map by placing our articulated model in the given initial pose. We "color" each limb of the articulated model with a unique identifier (a "limb number") and take a synthetic picture of the articulated model. This gives us a limb-assignment map for the model in the given initial pose. Assuming our model is a fair approximation to the true articulated figure being imaged, this assignment map will be a good approximation to the true limb-assignment map.

However, there will be differences between our model's shape and the imaged figure's shape. These differences will result in some of the pixels in the assignment map being incorrect. We need to identify these incorrectly labelled pixels but we do not need to remap them to the correct value. Instead, we remap them to "background" to remove them from the set of constraints that we are solving. Since we typically have many more constraints (2N) than unknowns (K+5), using a smaller set of constraints is better than including inaccurate constraints.

We use the sensed depth map, along with the model's depth map, to identify mislabelled figure pixels. Whenever the difference between these two depth maps is above a threshold, we relabel that pixel as "background", thereby removing it from the set of constraints.

4.2 Constrained limb-based cross correlation

On all frames except for the first frame, we do not know the current pose. Instead all we have is the pose in the previous frame. We will use constrained limb-based cross correlation (described here) to coarsely update the estiimated pose, so that it more closely approximates the pose in the current frame.

The constrained limb-based cross correlation uses the cross correlation between the time-*t* and time-(*t*+1) brightness and depth images. It is "limb-based" since we use the time-*t* limb-assignment map to determine the image support of the cross correlation for each limb. It is "constrained" since we use the articulated model and the time-*t* pose to select the candidate offsets and rotations within the time-(*t*+1) images.

We start with the torso. For the torso motion, we consider a small number of candidate translations in world coordinates X, Y, and Z. We select the one that provides the maximum cross correlation on the torso's support.

For all other limbs, we estimate their best rotations/ translations using chains of limbs. We create separate chains of limbs, starting from the torso. For our model we have four chains (two legs and two arms) with two links each (upper and lower arm/leg). We consider each chain separately, anchoring each one at the previously estimated torso offset.

We consider a small number of candidate articulation angles for each limb in the chain. We create chains of candidate articulations by combining across limbs. For each candidate articulation chain, we rotate and translate each limb's image segments according to the displacement of that limb under the candidate articulation chain. To avoid the overhead of image interpolation, we use zero-order hold to provide non-zero rotations of the limbs. We select the articulation chain that provides the maximim cross correlation on the combined supports of all the limbs in the chain.

This process allows us to coarsely update all the pose parameters except for the torso's rotation. We found that the torso rotation was slow enough between frames that updating that estimate was not required for creating reasonably accurate limb assignment map.

4.3 Creating subsequent limb-assignment maps

Throughout this paper, we assume that we start off kinowing the articulated pose of the figure in the first frame of the sequence. Since we know the true pose in the first image, creating an assignment map is easier than it will be in subsequent images, where we only have (estimates of) the articulated pose *in the previous frame*. Hence, creating the first assignment map is simpler than creating subsequent assignment maps.

We use the ideas in Sections 4.1 and 4.2 to create the limb-assignment maps for all the frames after the first frame. To do this, we first create an updated limb-assignment map for the previous frame, using the approach described in Section 4.1. In most cases, this improves the limb-assignment map for the previous frame, since it uses the pose estimate given be the solution to our previous frame's constraint equations (instead of the coarsely updated pose estimate that was previiously generated).

Once we have this assignment map for the previous frame, we use constrained limb-based cross correlation (Section 4.2) to coarsely estimate the current frame's pose from the previous frame's pose.

Finally, we use this coarsely updated pose estimate with the current frames in the approach described in Section 4.1. This gives us our limb-assignment map for the current frame.

5 Discrete-shift extension to constraint equations

The quality of the constraints from Section 3 also depends on the accuracy of the first-order Taylor-series expansion used in the BCCE (and ZCCE). This first-order approximation often fails on large motions. Classically, authors compensate for this failure by estimating the motion, warping the images according to that motion estimate and repeating. This iterative estimation approach has several drawbacks. It is computationally expensive, requiring sample interpolation of the image being warped, recomputation of its spatial derivatives, and multiple formulations and solutions of the constraint equations at each time step. It introduces interpolation errors, both in the warped image values and in the spatial derivatives. Finally, for large motions, the initial motion estimates may actually point away from the true solution.

We can improve the accuracy of BCCE (and ZCCE) without iteration, without interpolation, and without recomputing spatial derivatives. We do this by allowing the focus of expansion (FOE) of each pixel to shift “independently” by some integer amount, (S_x, S_y). Shifting the FOE by (S_x, S_y), the BCCE and ZCCE become:

$$-\begin{bmatrix} B(x-S_x, y-S_y, t+1) - B(x, y, t) \\ Z(x-S_x, y-S_y, t+1) - Z(x, y, t) \end{bmatrix} + S_x \begin{bmatrix} B_x(x, y, t) \\ Z_x(x, y, t) \end{bmatrix} + S_y \begin{bmatrix} B_y(x, y, t) \\ Z_y(x, y, t) \end{bmatrix} \approx \begin{bmatrix} B_x(x, y, t) & B_y(x, y, t) & 0 \\ Z_x(x, y, t) & Z_y(x, y, t) & -1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_Z \end{bmatrix} \qquad (12)$$

Equation (12) treats each constraint equation as if the t+1th frame translates rigidly by (S_x, S_y). As long as (S_x, S_y) is integer valued, we do not have to interpolate the image. Since our equations assume rigid translation, we do not need to recompute the spatial derivatives of the tth frame (as we would have to do, if we were warping the t+1th frame). The important thing to realize is that, even though each individual constraint derived from equation (12) acts as though the frame was rigidly translated, the set of constraints across the visible image does *not* have to share (S_x, S_y) values. Instead, at each pixel, we can select a new (S_x, S_y), according to what we expect will be a good shift for that pixel.

We now have the freedom of choosing a distinct value (S_x, S_y) for each visible point on the figure. We could do this within the traditional iterative motion estimation framework, using the (rounded) initial or previous motion estimates as the FOE for the next iteration.

In this paper, we instead use constrained limb-based cross correlations (Section 4.2) between times t and $t+1$ of the brightness and depth images to select the FOE. The constrained limb-based cross correlation takes the previous pose estimate and the previous support map along with the previous and current depth and color images and returns a (coarsely) updated pose estimate.

We derive an in-plane rotation and translation for each limb from this updated pose and the previous pose. We project the axis of each limb onto the image plane under the two poses. We then use the translation and rotation that best aligns the two axis projections for each limb.

Having selected a nominal translation/rotation for a limb, we use, for the FOE for each pixel on the limb, the integer-valued offset for S_x and S_y nearest to the offset dictated by the selected translation/rotation. We force S_x and S_y to be integer valued in order to reduce the computation and to avoid interpolation errors.

Equation (12), with equations (2), (5), (6), (7), (8) and (10), provides us with N BCCE and N ZCCE constraints, which we can solve with least squares, on $K+5$ unknowns ($p_0(t+1)$, $R_0(t+1)$, and $\theta_1(t+1)$ through $\theta_{K-1}(t+1)$). Once we have that solution, equation (11) provides the nonlinear mapping from ω_{v_0} to $R_0(t+1)$. In Section 7, we refer to these as “shifted” constraint equations.

6 Selecting Coordinate Systems

We must use camera-centric coordinates in the constraint equation (2). However, we can use any coordinate system that is offset by a known rotation/translation from the camera coordinate system for the coordinates (q_X, q_Y, q_Z) in equation (5). To improve the conditioning of the constraint equation, we choose the centroid of the visible figure as the origin of (q_X, q_Y, q_Z) [1]. For the “jump” sequence (described in Section 7), this coordinate translation reduces the condition number of the constraint matrix from 250 to 15.

We can also use any orientation/translation as our reference configuration $g_{sk}(0)$ in equation (3). The remainder of this section discusses how best to use this freedom to avoid estimation errors due to cross coupling between the body position and the body rotation estimates.

Equation (8) includes cross coupling between the estimates for $\dot{p}_0$ and ω_{v_0} according to the size of p_0. Previous derivations do not account for this coupling in their derivation: their estimation equations do not include $\begin{bmatrix} I & \hat{p}_0 \\ 0 & I \end{bmatrix}$. This coupling term is needed to correctly track a figure whose orientation relative to the camera changes. Yet, this coupling term also introduces a bias in our tracking estimates. In our simulations, when p_0 was non-zero and the figure was rotating relative to the camera, this resulted in a bias in the estimates of $\dot{p}_0$. For example, for the “jump” sequence, this bias accumulates over the sequence, increasing the location error by 4.5 times compared to the location error for the explicitly decoupled estimator, described here. The error increases linearly over time, so its effects become more detrimental as the sequence gets longer.

We avoid this bias by re-parameterizing our twists, at each time step, so that $p_0 = 0$. We can do this without affecting the coordinate-system origin for (q_X, q_Y, q_Z) by adjusting $g_{sk}(0)$, the reference configurations for the limbs (see equation (3)). This allows us to improve the conditioning of our constraints (as previously described) while still avoiding coupling.

To remove this coupling without altering the articulated figure's geometry, we need to subtract $\begin{bmatrix}(\omega_i \times R_0^T p_0)^T & 0^T\end{bmatrix}^T$ from each internal joint $\xi_i = \begin{bmatrix}v_i^T & \omega_i^T\end{bmatrix}^T$ $(i \geq 1)$.[1] This maintains the original geometry since, if $\exp\left(\begin{bmatrix}v_i \\ \omega_i\end{bmatrix}^{\wedge}\theta_i\right) = \begin{bmatrix}R_i & p_i \\ 0 & 1\end{bmatrix}$ then

$$e^{\begin{bmatrix}v_i\theta_i - (\omega_i\theta_i \times R_0^T p_0) \\ \omega_i\theta_i\end{bmatrix}^{\wedge}} = \begin{bmatrix}R_i & p_i + R_0^T p_0 - R_i R_0^T p_0 \\ 0 & 1\end{bmatrix}$$

$$= \begin{bmatrix}I & R_0^T p_0 \\ 0 & 1\end{bmatrix}\begin{bmatrix}R_i & p_i \\ 0 & 1\end{bmatrix}\begin{bmatrix}I & -R_0^T p_0 \\ 0 & 1\end{bmatrix}$$

so that

$$g_{s0} = \begin{bmatrix}R_0 & p_0 \\ 0 & 1\end{bmatrix} e^{\hat{\xi}_1\theta_1} \dots e^{\hat{\xi}_{k-1}\theta_{k-1}} g_{sk}(0)$$

$$= \left(\begin{bmatrix}R_0 & p_0 \\ 0 & 1\end{bmatrix}\begin{bmatrix}I & -R_0^T p_0 \\ 0 & 1\end{bmatrix}\right)\left(\begin{bmatrix}I & R_0^T p_0 \\ 0 & 1\end{bmatrix} e^{\hat{\xi}_1\theta_1}\begin{bmatrix}I & -R_0^T p_0 \\ 0 & 1\end{bmatrix}\right)$$

$$\dots\left(\begin{bmatrix}I & R_0^T p_0 \\ 0 & 1\end{bmatrix} e^{\hat{\xi}_k\theta_k}\begin{bmatrix}I & -R_0^T p_0 \\ 0 & 1\end{bmatrix}\right)\left(\begin{bmatrix}I & R_0^T p_0 \\ 0 & 1\end{bmatrix} g_{sb}(0)\right)$$

$$= \begin{bmatrix}R_0 & 0 \\ 0 & 1\end{bmatrix} e^{\begin{bmatrix}\xi_1\theta_1 - (\omega_1\theta_1 \times R_0^T p_0) \\ \omega_1\theta_1\end{bmatrix}^{\wedge}} \dots e^{\begin{bmatrix}\xi_{k-1}\theta_{k-1} - (\omega_{k-1}\theta_{k-1} \times R_0^T p_0) \\ \omega_{k-1}\theta_{k-1}\end{bmatrix}^{\wedge}}$$

$$\left(g_{sb}(0) + \begin{bmatrix}0 & R_0^T p_0 \\ 0 & 0\end{bmatrix}\right)$$

When we use these transformations to remove the cross-coupling between $\dot{p}_0$ and ω_{v_0}, we also need to transform $\dot{p}_0$ back to the original coordinate system. We do this by setting $p_0(t+1) = \dot{p}_0 + R_0(t+1)R_0^T(t)p_0(t)$.

7 Results

7.1 Tracking results on synthetic sequences

Synthetic image sequences provide ground truth that allow us to quantitatively analyze our techniques. We generated four synthetic sequences. The two "jump" sequences, one "slow" and one "fast", are 200 frames long and show the figure jumping by spreading his upper legs (thighs) and bending his lower legs (calves) up behind him while also raising his upper arms and bending his lower arms forward. During the "jump" sequences, the camera rotates around the figure, starting near 60° to his left and ending near 60° to his right. In the "slow jump" sequence, the camera and each joint move about 0.6 degrees between frames; in the "fast jump" sequence, the motion is about 3 degrees between frames. The two "walk" sequences, one "slow" and one "fast", are 120 frames long. The "slow walk" sequence shows one full walk cycle; the "fast walk" shows five full walk cycles. Both have the camera at 60° to the figure's right. In all four sequences, the figure is created from 10 ellipsoidal cylinders with three twist axes between each cylinder (allowing full 3-DOF rotation between limbs). In all four sequences, the camera is two body lengths from the figure, resulting in noticeable perspective effects. For the brightness images, a plaid pattern was texture mapped onto each of the cylinders. The depth maps show the smooth ellipsoidal cylinders.

We use these four sequences to evaluate the robustness of BCCE using estimated depth ("BCCE-only"), of BCCE using true depth ("BCCE+depth"), of ZCCE alone ("ZCCE-only"), and of BCCE and ZCCE together ("BCCE+ZCCE"). We also examine the performance improvements gotten by shifting the FOE, as described in Section 5. All of our tracking results are shown in movies on our web site: http://web.interval.com/papers/1999-122/.

Our implementation of tracking with BCCE-only is similar to [3], with the addition of the machinery of Section 5. True depth values are not used in the BCCE-only case; instead, as in [3], depth values are estimated by positioning the articulated figure according to the current pose estimate and computing the corresponding depths. Our simulations show that BCCE-only, without true depth, tends to fail after extended periods of tracking. In the "slow jump" sequence, between the 20th and 60th frames, the estimated pose slowly quickly rotates around its vertical axis until it is facing backwards (Figure 1). In the "slow walk" sequence, the estimated pose completely loses track of the figure by the 17th frame (Figure 2).

In contrast, when we added the true depth images, the body tracking behavior stabilized on the "slow jump" and "slow walk" sequences, even using BCCE without the ZCCE (see Figures 3 and 4). With BCCE+depth, the tracking mistakes tend to be in terms of depth. The most obvious example is the left (rear) arm of the "walk" sequence (the BCCE+depth estimate for the arm "floats" away from the camera). With ZCCE-only, the tracking errors tend to concentrate more in the image plane. Finally, combining

[1] We also implicitly add an offset of $R_0^T p_0$ to the last column of $g_{sk}(0)$; however, since $g_{sk}(0)$ never appears in our final constraint equations, this offset is more conceptual than computational.

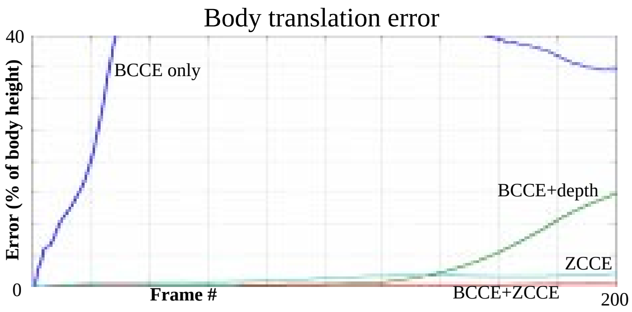


Figure 1: Body translation error on the “slow jump” sequence.

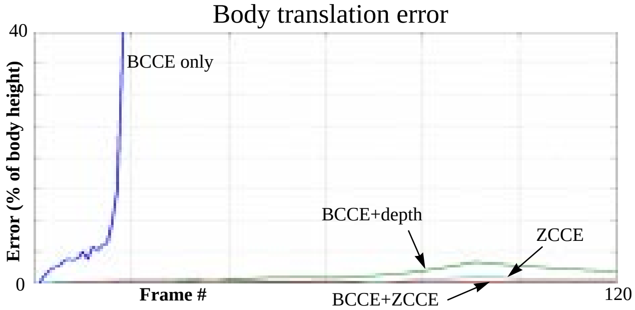


Figure 2: Body translation error on the “slow walk” sequence.

BCCE and ZCCE drastically reduces the errors seen in the separate estimation sequences.

We used our “fast” sequences to examine the performance of the unshifted and shifted BCCE+depth, ZCCE-only, and BCCE+ZCCE, under more demanding circumstances than those provided by the “slow” sequences. We did not include the BCCE-only solutions in this more stringent test, since BCCE-only already fails rapidly on the simpler “slow” sequences. On the “fast” sequences, both shifted and unshifted constraint equations track the body translation and rotation, with varying levels of accuracy but none of our *unshifted* constraint equations can track the limb motions. When we shift the FOE, the tracking is more robust. The improved tracking behavior is apparent from Figures 7 and 8.

As shown in Figures 7 and 8, adding shifted-FOE tracking is obviously better. However, even with the shifted-FOE, the tracking error is increasing over the course of the sequence. This is due to the fact that our pose estimation process is purely differential, so that tracking errors tend to accumulate from one frame to the next. Even with this shortcoming, our tracking behavior is quite good.

7.2 Tracking on a real data sequence

We tracked the motion of a person walking parallel to the image plane through a 21-frame sequence. The depth data came from a calibrated stereo camera pair, mounted at ceiling level. The disparities were estimated using the approach outlined by Woodfill [13]. We filled in small holes created by inconsistent depth measures using simple morphological operations.

We used the full-color video images (as well as the depth images) to form the limb-assignment maps but, for simplicity of coding, we used only the grayscale values in the BCCE itself. For this real-data sequence, we solved our linear constraint equations under the constraint that the weighted L_∞ of the solution vector be less than or equal to preset limits. We limited the motion in the 1/15 second between frames to: 0.1 m torso motion in X, Y, or Z; 3º, 12º, and 1º of torso tilt around the torso’s X, Y, and Z axes; 17º, 0º, and 6º of rotation around the upper legs’ X, Y, and Z axes; and 29º, 0º, and 6º of rotation about the lower legs’ X,Y, and Z axes.

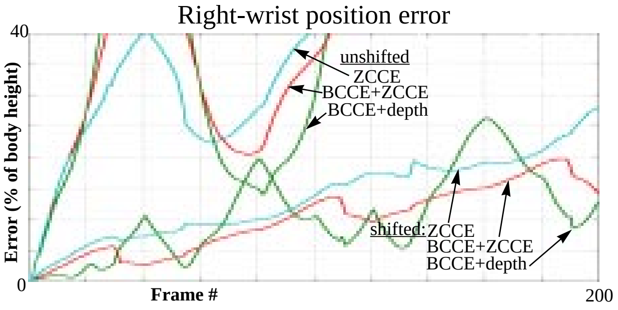


Figure 3: BCCE+ZCCE tracking error for the “fast jump” right wrist. Errors were computed after having removed the body-position error.

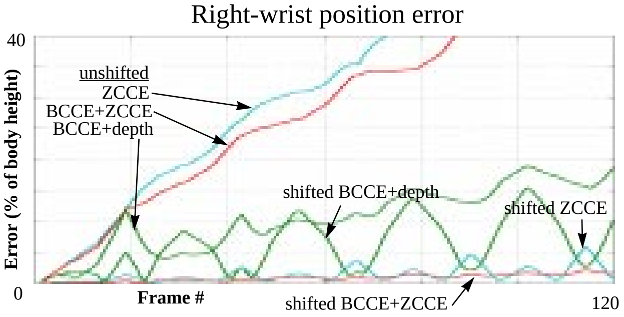
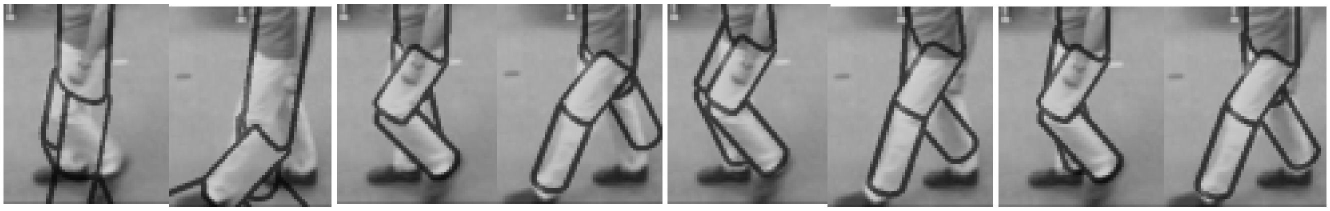


Figure 4: BCCE+ZCCE tracking error for the “fast walk” right wrist. Errors were computed after having removed the body-position error.

Figure 5 shows example frames from our tracking experiments. Movies showing our results are on http://web.interval.com/papers/1999-122/.

Again, the BCCE-only approach provided the least stable tracking results. To a large extent this is due to misassignment of limb and background pixels. BCCE-only does not have the depth map available to help it distinguish the background pixels from the leg pixels or the front leg from the back leg pixels. As a result, the leg motions are not tracked. The torso’s forward motion is well tracked, but the torso’s vertical motion is grossly misestimated, with the model sliding down to rest on the legs. This misestimation is probably due to the simultaneous, coupled solution for the torso and leg motions: the errors in the leg-motion constraints affect the torso-motion estimates. Since the estimation techniqe is purely differential, this torso-motion error accumulates over the length of the sequence.

The other three tracking approaches track the leg motions well. The largest tracking errors occur when the back leg is occluded. Again, with the accumulation of error due to our differential approach, none of the tracking meth-



BCCE-only BCCE+depth ZCCE-only BCCE+ZCCE

Figure 5: Real tracking results. The overlay shows the outline of the articulated-model limbs in the pose estimated for each frame. The images shown here have been cropped to show only the walking figure. The 11^{th} and 15^{th} frames from the sequence are shown.

ods fully recover from the error, especially in tracking the back upper leg. However, all of them continue to track the back lower leg, even when the articulated model is only partially above the back leg. This robustness comes from the depth map: we have removed many pixel assignments that would otherwise be incorrect using the depth-mismatch thresholds.

8 Conclusions

We have combined BCCE and ZCCE with twist mathematics to create over-constrained systems of equations for estimating the position, rotation and joint angles of an articulated figure. As long as depth images are available, these constraints are linear and can be solved simply and efficiently using least squares. The only non-linear relation (equation (11)) acts as an auxiliary equation and is used at each time step but only after the linear least-squares solution has been found. By including this non-linear auxiliary equation, we avoid having to introduce a first-order Taylor-series approximation to the body rotation matrix.

Our simulation results argue strongly for video-rate depth information, from either stereo cameras or other sensors. Without the true depth information, the tracking behavior became unstable and failed catastrophically within sixty frames of our test sequences.

In contrast, we obtained good tracking results on our sequences using true depth data. The results for the combined BCCE and ZCCE are better than the results for either set of equations alone. The accuracy improves when BCCE is added since our brightness images have higher spatial frequencies than do our depth images, thereby providing tighter constraints on the figure's X/Y motions. The accuracy improves when ZCCE is added since it is a tighter constraint on the figure's Z motions than we can otherwise infer from brightness-image perspective effects. Also, the ZCCE is much less sensitive than the BCCE to illumination and reflectance changes.

On our "slow" sequences, as long as we used the true depth values, we did not need to shift our FOE to get good tracking results. However, on our "fast" sequences, we did need the shifted FOE. Without the shift, the pose estimates lose track of the different limbs and can not recover the correct alignment. Unlike previously proposed iterative estimates, our proposed method for shifting the FOE does not require iteration or image interpolation. Currently we select the FOEs based on the correlation peak between the two images. We only consider a small number of potential rotations/translations, so the computational requirements for this correlation are not high.

One area that needs to be investigated further is how to avoid error accumulation in our pose estimates. Our current constraints are purely differential and therefore are only marginally stable. We need to interleave this differential method with a non-differential technique (such as "generate-and-test"). Without having a non-differential technique, the error will grow until the tracking fails catastrophically. Our differential method can improve the local accuracy and sensitivity of a non-differential technique while the non-differential technique will make our differential approach unconditionally stable.

References

[1] Harville, Rahimi, Darrell, Gordon, Woodfill, "3D pose tracking with linear depth and brightness constraints," *ICCV*, vol. 1 pp. 206–213, Sep 1999.

[2] Murray, Li, Sastry, *Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Baton FL, 1994.

[3] Bregler, Malik, "Tracking People with Twists and Exponential Maps," *CVPR*, pp. 8–15, Santa Barbara, CA, June 1998.

[4] Aggarwal, Cai, Liao, Sabata, "Nonrigid Motion Analysis: Articulated and Elastic Motion," *Computer Vision and Image Understanding* 70(2):142–156, May 1998.

[5] Pentland, Horowitz, "Recovery of Nonrigid Motion and Structure," PAMI 13(7):730–742, July 1991.

[6] Yamamoto, Sato, Kawada, Kondo, Osaki, "Incremental Tracking of Human Actions from Multiple Views," *CVPR*, pp. 2–7, June 1998.

[7] Lin, "Tracking Articulated Objects in Real-Time Range Image Sequences," *ICCV*, vol. 1 pp. 648–653, Sep 1999.

[8] Wren, Pentland, "Dynamic models of human motion," *International Conference on Automatic Face and Gesture Recognition*, pp. 22-27, April 1998.

[9] Kakadiaris, Metaxas, Bajcsy, "Active part-decomposition, shape and motion estimation of articulated objects: a physics-based approach," *CVPR*, pp. 980-984, June 1994.

[10] Gavrila, Davis, "3-D model-based tracking of humans in action: a multi-view approach," *CVPR*, pp. 73-80, June 1996.

[11] Ju, Black, Yacoob, "Cardboard people: a parameterized model of articulated image motion," *International Conference on Automatic Face and Gesture Recognition*, pp. 38-44, Oct. 1996.

[12] Cham, Rehg, "A multiple hypothesis approach to figure tracking," *CVPR*, pp. 239-244, June 1999.

[13] Woodfill, Van Herzen, "Real-time stereo vision on the PARTS reconfigurable computer," Symposium on Field-Programmable Custom Computing Machines, pp. 242–250, April 1997.