Tracking People with Twists and Exponential Maps

Christoph Bregler and Jitendra Malik
Computer Science Division
U.C. Berkeley
Berkeley, CA 94720
{bregler,malik}@cs.berkeley.edu
http://www.cs.berkeley.edu/~bregler/digmuy.html

Abstract

This paper demonstrates a new visual motion estimation technique that is able to recover high degree-of-freedom articulated human body configurations in complex video sequences. We introduce the use of a novel mathematical technique, the product of exponential maps and twist motions, and its integration into a differential motion estimation. This results in solving simple linear systems, and enables us to recover robustly the kinematic degrees-of-freedom in noise and complex self occluded configurations. We demonstrate this on several image sequences of people doing articulated full body movements, and visualize the results in re-animating an artificial 3D human model. We are also able to recover and re-animate the famous movements of Eadweard Muybridge's motion studies from the last century. To the best of our knowledge, this is the first computer vision based system that is able to process such challenging footage and recover complex motions with such high accuracy.

1 Introduction

The estimation of image motion without any domain constraints is an underconstrained problem. Therefore all proposed motion estimation algorithms involve additional constraints about the assumed motion structure. One class of motion estimation techniques are based on parametric algorithms [3]. These techniques rely pm solving a highly overconstrained system of linear equations. For example, if an image patch could be modeled as a planar surface, an affine motion model with low degrees of freedom (6 DOF) can be estimated. Measurements over many pixel locations have to comply with this motion model. Noise in image features and ambiguous motion patterns can be overcome by measurements from features at other image locations. If the motion can be approximated by this simple motion model, sub-pixel accuracy can be achieved.

Problems occur if the motion of such a patch is not well described by the assumed motion model. Others have shown how to extend this approach to multiple independent moving motion areas [15, 1, 26]. For each area, this approach still has the advantage that a large number of measurements are incorporated into a low DOF linear motion estimation. Problems occur if some of the areas do not have a large number of pixel locations or have mostly noisy or ambiguous motion measurements. One example is the measurement of human body motion. Each body segment can be approximated by one rigid moving object. Unfortunately, in standard video sequences the area of such body segments are very small, the motion of leg and arm segments is ambiguous in certain directions (for example parallel to the boundaries), and deforming clothes cause noisy measurements.

If we increase the ratio between the number of measurements and the degrees of freedom, the motion estimation will be more robust. This can be done using additional constraints. Body segments don't move independently; they are attached by body joints. This reduces the number of free parameters dramatically. A convenient way of describing these additional domain constraints is the *twist* and *product of exponential map* formalism for kinematic chains [20]. The motion of one body segment can be described as the motion of the previous segment in a kinematic chain and an angular motion around a body joint. This adds just a single DOF for each additional segment in the chain. In addition, the exponential map formulation makes it possible to relate the image motion vectors linearly to the angular velocity.

Others have modeled the human body with rigid segments connected at joints [14, 24, 23, 13, 10, 9, 16, 17], but use different representations and features (for example Denavit-Hartenburg and edge detection). The introduction of twists and product of exponential maps into region-based motion estimation simplifies the estimation dramatically and leads to robust tracking results. Besides tracking, we also outline how to fine-tune the kinematic model itself. Here the ratio between the number of measurements and the degrees of freedom is even larger, because we can optimize over a complete image sequence.

Alternative solutions to tracking of human bodies were proposed by [27] in tracking color blobs, and by [11] in using motion templates. Nonrigid models were proposed by [22, 7, 5, 6].

Section 2 introduces the new motion tracking framework and its mathematical formulation, section 3 details our experiments, and we discuss the results and future directions in section 4.

2 Motion Estimation

We first describe a commonly used region-based motion estimation framework [3, 25], and then describe the extension to kinematic chain constraints [20].

2.1 Preliminaries

Assuming that changes in image intensity are only due to translation of local image intensity, a parametric image motion between consecutive time frames t and $t+1$ can be described by the following equation:

$$I(x+\mathbf{u}_x(x,y,\phi), y+\mathbf{u}_y(x,y,\phi), t+1) = I(x,y,t) \quad (1)$$

$I(x,y,t)$ is the image intensity. The motion model $\mathbf{u}(x,y,\phi) = [\mathbf{u}_x(x,y,\phi), \mathbf{u}_y(x,y,\phi)]^T$ describes the pixel displacement dependent on location (x,y) and model parameters ϕ. For example, a 2D affine motion model with parameters $\phi = [a_1, a_2, a_3, a_4, d_x, d_y]^T$ is defined as

$$\mathbf{u}(x,y,\phi) = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix} \quad (2)$$

The first-order Taylor series expansion of (1) leads to the commonly used gradient formulation [18]:

$$I_t(x,y) + [I_x(x,y), I_y(x,y)] \cdot \mathbf{u}(x,y,\phi) = 0 \quad (3)$$

$I_t(x,y)$ is the temporal image gradient and $[I_x(x,y), I_y(x,y)]$ is the spatial image gradient at location (x,y). Assuming a motion model of K degrees of freedom (in case of the affine model $K=6$) and a region of $N > K$ pixels, we can write an over-constrained set of N equations. For the case that the motion model is linear (as in the affine case), we can write the set of equations in matrix form (see [3] for details):

$$\mathbf{H} \cdot \phi + \vec{z} = \vec{0} \quad (4)$$

where $\mathbf{H} \in \Re^{N \times K}$, and $\vec{z} \in \Re^N$. The least squares solution to (3) is:

$$\phi = -\left(\mathbf{H}^T \cdot \mathbf{H}\right)^{-1} \cdot \mathbf{H}^T \vec{z} \quad (5)$$

Because (4) is the first-order Taylor series linearization of (1), we linearize around the new solution and iterate. This is done by warping the image $I(t+1)$ using the motion model parameters ϕ found by (5). Based on the re-warped image we compute the new image gradients (3). Repeating this process is equivalent to a Newton-Raphson style minimization.

A convenient representation of the shape of an image region is a probability mask $w(x,y) \in [0,1]$. $w(x,y) = 1$ declares that pixel (x,y) is part of the region. Equation (5) can be modified, such that it weights the contribution of pixel location (x,y) according to $w(x,y)$:

$$\phi = -\left((\mathbf{W} \cdot \mathbf{H})^T \cdot \mathbf{H}\right)^{-1} \cdot (\mathbf{W} \cdot \mathbf{H})^T \vec{z} \quad (6)$$

$\mathbf{W}$ is an $N \times N$ diagonal matrix, with $\mathbf{W}(i,i) = w(x_i, y_i)$. We assume for now that we know the exact shape of the region. For example, if we want to estimate the motion parameters for a human body part, we supply a weight matrix $\mathbf{W}$ that defines the image support map of that specific body part, and run this estimation technique for several iterations. Section 2.4 describes how we can estimate the shape of the support maps as well.

Tracking over multiple frames can be achieved by applying this optimization technique successively over the complete image sequence.

2.2 Twists and the Product of Exponential Formula

In the following we develop a motion model $\mathbf{u}(x,y,\phi)$ for a 3D kinematic chain under scaled orthographic projection and show how these domain constraints can be incorporated into one linear system similar to (6). ϕ will represent the 3D pose and angle configuration of such a kinematic chain and can be tracked in the same fashion as already outlined for simpler motion models.

2.2.1 3D pose

The pose of an object relative to the camera frame can be represented as a rigid body transformation in $\Re^3$ using homogeneous coordinates (we will use the notation from [20]):

$$q_c = \mathbf{G} \cdot q_o \quad \text{with} \quad \mathbf{G} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & d_x \\ r_{2,1} & r_{2,2} & r_{2,3} & d_y \\ r_{3,1} & r_{3,2} & r_{3,3} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$q_o = [x_o, y_o, z_o, 1]^T$ is a point in the object frame and $q_c = [x_c, y_c, z_c, 1]^T$ is the corresponding point in the camera frame. Using scaled orthographic projection with scale s, the point q_c in the camera frame gets projected into the image point $[x_{im}, y_{im}]^T = s \cdot [x_c, y_c]^T$.

The 3D translation $[d_x, d_y, d_z]^T$ can be arbitrary, but the rotation matrix:

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} \in SO(3) \quad (8)$$

has only 3 degrees of freedom. Therefore the rigid body transformation $\mathbf{G} \in SE(3)$ has a total of 6 degrees of freedom.

Our goal is to find a model of the image motion that is parameterized by 6 degrees of freedom for the 3D rigid motion and the scale factor s for scaled orthographic projection. *Euler angles* are commonly used to constrain the rotation matrix to $SO(3)$, but they suffer from singularities and don't lead to a simple formulation in the optimization procedure (for example [2] propose a 3D ellipsoidal tracker based on Euler angles). In contrast, the *twist* representation provides a more elegant solution [20] and leads to a very simple linear representation of the motion model. It is based on the observation that every rigid motion can be represented as a rotation around a 3D axis and a translation along this axis. A twist ξ has two representations: (a) a 6D vector, or (b) a 4×4 matrix with the upper 3×3 component as a skew-symmetric matrix:

$$\xi = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad \text{or} \quad \hat{\xi} = \begin{bmatrix} 0 & -\omega_z & \omega_y & v_1 \\ \omega_z & 0 & -\omega_x & v_2 \\ -\omega_y & \omega_x & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (9)$$

ω is a 3D unit vector that points in the direction of the rotation axis. The amount of rotation is specified with a scalar angle θ that is multiplied by the twist: $\xi\theta$. The v component determines the location of the rotation axis and the amount of translation along this axis. See [20] for a detailed geometric interpretation. It is convenient to drop the θ coefficient by relaxing the constraint that ω is unit length. Therefore $\xi \in \Re^6$.

It can be shown [20] that for any arbitrary $\mathbf{G} \in SE(3)$ there exists a $\xi \in \Re^6$ twist representation.

A twist can be converted into the $\mathbf{G}$ representation with following exponential map:

$$\mathbf{G} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & d_x \\ r_{2,1} & r_{2,2} & r_{2,3} & d_y \\ r_{3,1} & r_{3,2} & r_{3,3} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \mathbf{e}^{\hat{\xi}} = \mathbf{I} + \hat{\xi} + \frac{(\hat{\xi})^2}{2!} + \frac{(\hat{\xi})^3}{3!} + \ldots \tag{10}$$

2.2.2 Twist motion model

At this point we would like to track the 3D pose of a rigid object under scaled orthographic projection. We will extend this formulation in the next section to a kinematic chain representation. The pose of an object is defined as $[s, \xi^T]^T = [s, v_1, v_2, v_3, \omega_x, \omega_y, \omega_z]^T$. A point q_o in the object frame is projected to the image location (x_{im}, y_{im}) with:

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot s \cdot \mathbf{e}^{\hat{\xi}} \cdot q_o \tag{11}$$

The image motion of point (x_{im}, y_{im}) from time t to time $t+1$ is:

$$\begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} = \begin{bmatrix} x_{im}(t+1) - x_{im}(t) \\ y_{im}(t+1) - y_{im}(t) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \left(s(t+1) \cdot \mathbf{e}^{\hat{\xi}(t+1)} \cdot q_o - s(t) \cdot \mathbf{e}^{\hat{\xi}(t)} \cdot q_o \right)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \left((1+s') \cdot \mathbf{e}^{\hat{\xi}'} - \mathbf{I} \right) \cdot s(t) q_c \tag{12}$$

$$\text{with} \quad \begin{aligned} \hat{\xi}(t+1) &= \hat{\xi}(t) + \hat{\xi}' \\ s(t+1) &= s(t) \cdot (1+s') \end{aligned}$$

Using the first order Taylor expansion from (10) we can approximate:

$$(1+s') \cdot \mathbf{e}^{\hat{\xi}} \approx (1+s') \cdot \mathbf{I} + (1+s') \cdot \hat{\xi} \tag{13}$$

and can rewrite (12) as:

$$\begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} = \begin{bmatrix} s' & -\omega'_z & \omega'_y & v'_1 \\ \omega'_z & s' & -\omega'_x & v'_2 \end{bmatrix} \cdot q_c \tag{14}$$

with

$$\omega(t+1) = \omega(t) + \frac{1}{1+s'} \cdot \omega'$$

$$v(t+1) = v(t) + \frac{1}{1+s'} \cdot v'$$

$\phi = [s', v'_1, v'_2, \omega'_x, \omega'_y, \omega'_z]^T$ codes the relative scale and twist motion from time t to $t+1$. Note that (14) does not include v'_3. Translation in the Z direction of the camera frame is not measurable under scaled orthographic projection.

Equation (14) describes the image motion of a point (x_i, y_i) in terms of the motion parameters ϕ and the corresponding 3D point $q_c(i)$ in the camera frame. The 3D point $q_c(i)$ is computed by intersecting the camera ray of the image point (x_i, y_i) with the 3D model. In this paper we assume that the body segments can be approximated by ellipsoidal 3D blobs. Therefore q_c is the solution of a quadratic equation. This computation has to be done only once for each new image. It is outside the Newton-Raphson iterations. It could be replaced by more complex models and rendering algorithms.

Inserting (14) into (3) leads to:

$$I_t + I_x \cdot [s', -\omega'_z, \omega'_y, v'_1] \cdot q_c + I_y \cdot [\omega'_z, s', -\omega'_x, v'_2] \cdot q_c = 0$$

$$\Leftrightarrow \qquad I_t(i) + H_i \cdot [s, v'_1, v'_2, \omega'_x, \omega'_y, \omega'_z]^T = 0 \tag{15}$$

with $I_t := I_t(x_i, y_i), I_x := I_x(x_i, y_i), I_y := I_y(x_i, y_i)$

For N pixel positions we have N equations of the form (15). This can be written in matrix form:

$$\mathbf{H} \cdot \phi + \vec{z} = \mathbf{0} \tag{16}$$

with

$$\mathbf{H} = \begin{bmatrix} H_1 \\ H_2 \\ \ldots \\ H_N \end{bmatrix} \quad \text{and} \quad \vec{z} = \begin{bmatrix} I_t(x_1, y_1) \\ I_t(x_2, y_2) \\ \ldots \\ I_t(x_N, y_N) \end{bmatrix}$$

Finding the least-squares solution (3D twist motion ϕ) for this equation is done using (6).

2.2.3 Kinematic chain as a Product of Exponentials

So far we have parameterized the 3D pose and motion of a body segment by the 6 parameters of a twist ξ. Points on this body segment in a canonical object frame are transformed into a camera frame by the mapping $\mathbf{G_0} = e^{\hat{\xi}}$. Assume that a second body segment is attached to the first segment with a joint. The joint can be defined by an axis of rotation in the object frame. We define this rotation axis in the object frame by a 3D unit vector ω_1 along the axis, and a point q_1 on the axis (figure 1). This is a revolute joint, and can be modeled by a twist ([20]):

$$\xi_1 = \begin{bmatrix} -\omega_1 \times q_1 \\ \omega_1 \end{bmatrix} \tag{17}$$

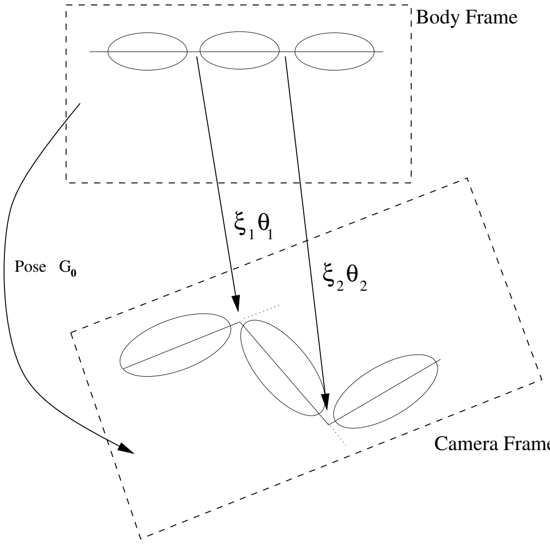

Figure 1: Kinematic chain defined by twists

A rotation of angle θ_1 around this axis can be written as:

$$\mathbf{g_1} = e^{\hat{\xi}_1 \cdot \theta_1} \tag{18}$$
$$\tag{19}$$

The global mapping from object frame points on the first body segment into the camera frame is described by the following product:

$$\begin{aligned} \mathbf{g}(\theta_1) &= \mathbf{G_0} \cdot e^{\hat{\xi}_1 \cdot \theta_1} \quad (20) \\ q_c &= \mathbf{g}(\theta_1) \cdot q_o \end{aligned}$$

If we have a chain of $K+1$ segments linked with K joints (kinematic chain) and describe each joint by a twist ξ_k, a point on segment k is mapped from the object frame into the camera frame dependent on $\mathbf{G_0}$ and angles θ_1, θ_2, ..., θ_k:

$$\mathbf{g_k}(\theta_1, \theta_2, ..., \theta_k) = \mathbf{G_0} \cdot e^{\hat{\xi}_1 \cdot \theta_1} \cdot e^{\hat{\xi}_2 \cdot \theta_2} \cdot \ldots \cdot e^{\hat{\xi}_k \cdot \theta_k} \tag{21}$$

This is called the **product of exponential maps** for kinematic chains.

The velocity of a segment k can be described with a twist V_k that is a linear combination of twists $\xi'_1, \xi'_2, ..., \xi'_k$ and the angular velocities $\dot{\theta}_1, \dot{\theta}_2, ..., \dot{\theta}_k$ (see [20] for the derivations):

$$\begin{aligned} V_k &= \xi'_1 \cdot \dot{\theta}_1 + \xi'_2 \cdot \dot{\theta}_2 + \ldots \xi'_k \cdot \dot{\theta}_k \quad (22) \\ \xi'_k &= \mathbf{Ad}_{e^{\hat{\xi}_1 \theta_1} \cdot \ldots \cdot e^{\widehat{\xi_{k-1}} \theta_{k-1}}} \xi_k \end{aligned}$$

$\mathbf{Ad}_g$ is the adjoint transformation associated with g.[1]

Given a point q_c on the k'th segment of a kinematic chain, its motion vector in the image is related to the angular velocities by:

$$\begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \left[\hat{\xi}'_1 \cdot \dot{\theta}_1 + \hat{\xi}'_2 \cdot \dot{\theta}_2 + \ldots + \hat{\xi}'_k \cdot \dot{\theta}_k \right] \cdot q_c \tag{23}$$

Recall (15) relates the image motion of a point q_c to changes in pose $\mathbf{G}_0$. We combine (15) and (23) to relate the image motion to the combined vector of pose change and angular change $\mathbf{\Phi} = [s', v'_1, v'_2, \omega'_x, \omega'_y, \omega'_z, \dot{\phi}_1, \dot{\phi}_2, ..., \dot{\phi}_K]^T$:

$$I_t + H_i \cdot [s, v'_1, v'_2, \omega'_x, \omega'_y, \omega'_z]^T + J_i \cdot [\dot{\theta}_1, \dot{\theta}_2, \ldots \dot{\theta}_K]^T = 0 \tag{24}$$

$$[\mathbf{H}, \mathbf{J}] \cdot \mathbf{\Phi} + \vec{z} = \mathbf{0} \tag{25}$$

with

$$\mathbf{J} = \begin{bmatrix} J_1 \\ J_2 \\ \ldots \\ J_N \end{bmatrix} \qquad \text{and } \mathbf{H}, \vec{z} \text{ as before}$$

$$J_i = [J_{i1}, J_{i2}, ..., J_{iK}]$$

$$J_{ik} = \begin{cases} [I_x, I_y] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \hat{\xi}_k \cdot q_c & \\ 0 & \text{if pixel } i \text{ is on a segment that} \\ & \text{is not affected by joint } \xi_k \end{cases}$$

The least squares solution to (25) is:

$$\mathbf{\Phi} = -([\mathbf{H}, \mathbf{J}]^T \cdot [\mathbf{H}, \mathbf{J}])^{-1} \cdot [\mathbf{H}, \mathbf{J}]^T \cdot \vec{z} \tag{26}$$

$\mathbf{\Phi}$ is the new estimate of the pose and angular change between two consecutive images. As outlined earlier, this solution is based on the assumption that the local image intensity variations can be approximated by the first-order Taylor expansion (3). We linearize around this new solution and iterate. This is done in warping the image $I(t+1)$ using the solution $\mathbf{\Phi}$. Based on the re-warped image we compute the new image gradients. Repeating this process of warping and solving (26) is equivalent to a Newton-Raphson style minimization.

2.3 Multiple Camera Views

In cases where we have access to multiple synchronized cameras, we can couple the different views in one equation system. Let's assume we have C different camera views at the same time. View c corresponds to following equation system (from (25)):

$$[\mathbf{H_c}, \mathbf{J_c}] \cdot \begin{bmatrix} \mathbf{\Omega}_c \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \ldots \\ \dot{\phi}_K \end{bmatrix} + \vec{z_c} = \mathbf{0} \tag{27}$$

$\mathbf{\Omega}_c = [s'_c, v'_{1,c}, v'_{2,c}, \omega'_{x,c}, \omega'_{y,c}, \omega'_{z,c}]^T$ describes the pose seen from view c. All views share the same angular parameters, because the cameras are triggered at the same

[1] $\mathbf{Ad}_g = \begin{bmatrix} R & \hat{p} \cdot R \\ \mathbf{0} & R \end{bmatrix}$, and $g = \begin{bmatrix} R & p \\ 000 & 1 \end{bmatrix}$

time. We can simply combine all C equation systems into one large equation system:

$$\begin{bmatrix} \mathbf{H_1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{J_1} \\ \mathbf{0} & \mathbf{H_2} & \dots & \mathbf{0} & \mathbf{J_2} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{H_C} & \mathbf{J_C} \end{bmatrix} \cdot \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \dots \\ \Omega_C \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dots \\ \dot{\phi}_K \end{bmatrix} + \begin{bmatrix} \vec{z_1} \\ \vec{z_2} \\ \dots \\ \vec{z_C} \end{bmatrix} = \mathbf{0} \tag{28}$$

Operating with multiple views has three main advantages. The estimation of the angular parameters is more robust because (1) the number of measurements and therefore the number of equations increases with the number of views, (2) some angular configurations might be close to a singular pose in one view, whereas they can be estimated in a orthogonal view much better. (3) With more camera views, the chance decreases that one body part is occluded in all views.

2.4 Adaptive Support Maps using EM

As in (3), the update can be constrained to estimate the motion only in a weighted support map $\mathbf{W}_k$ for each segment k using:

$$\Phi = -\left((\mathbf{W}_k \cdot [\mathbf{H}, \mathbf{J}])^T \cdot [\mathbf{H}, \mathbf{J}] \right)^{-1} \cdot (\mathbf{W}_k \cdot [\mathbf{H}, \mathbf{J}])^T \vec{z} \tag{29}$$

We approximate the shape of the body segments as ellipsoids, and can compute the support map as the projection of the ellipsoids into the image. Such a support map usually covers a larger region, including pixels from the environment. That distracts the exact motion measurement. Robust statistics would be one solution to this problem [4]. Another solution is an EM-based layered representation [12, 15]. It is beyond the scope of this paper to describe this method in detail, but we would like to outline the method briefly: We start with an initial guess of the support map (ellipsoidal projection in this case). Given the initial $\mathbf{W}_k$, we compute the motion estimate Φ (M-step). Given such a Φ we can compute for each pixel location the probability that it complies with the motion model defined by Φ. We do this for each blob and the background (dominant motion) and normalize the sum of all probabilities per pixel location to 1. This results in new $\mathbf{W}_k$ maps that are better "tuned" to the real shape of the body segment. In this paper we repeat the EM iteration only once.

2.5 Tracking Recipe

We summarize the algorithm for tracking the pose and angles of a kinematic chain in an image sequence:

- **Input:** $I(t), I(t+1), \mathbf{G_0(t)}, \theta_1(t), \theta_2(t), ..., \theta_K(t)$ (Two images and the pose and angles for the first image).
- **Output:** $\mathbf{G_0(t+1)}, \theta_1(t+1), \theta_2(t+1), ..., \theta_K(t+1)$. (Pose and angles for second image).

1. Compute for each image location (x_i, y_i) in $I(t)$ the 3D point $q_c(i)$ (using ellipsoids or more complex models and rendering algorithm).
2. Compute for each body segment the support map W_k.
3. Set $\mathbf{G}_0(t+1) := \mathbf{G}_0(t)$, $\forall k : \theta_k(t+1) := \theta_k(t)$.
4. Iterate:
   (a) Compute spatiotemporal image gradients: I_t, I_x, I_y.
   (b) Estimate Φ using (29)
   (c) Update $G_0(t+1) := G_0(t+1) \cdot (1+s') \cdot e^{\frac{\hat{\xi}'}{1+s'}}$
   (d) $\forall k$ Update $\theta_k(t+1) := \theta_k(t+1) + \dot{\theta}_k$.
   (e) $\forall k$ Warp the region inside W_k of $I(t+1)$ by $\mathbf{G}_0(t+1) \cdot g_k(t+1) \cdot (\mathbf{G}(t) \cdot g_k(t))^{-1}$.

2.6 Initialization

The visual tracking is based on an initialized first frame. We have to know the initial pose and the initial angular configuration. If more than one view is available, all views for the first time step have to be known. A user clicks on the 2D joint locations in all views at the first time step. Given that, the 3D pose and the image projection of the matching angular configuration is found by minimizing the sum of squared differences between the projected model joint locations and the user supplied model joint locations. The optimization is done over the poses, angles, and body dimensions. Example body dimensions are "upper-leg-length", "lower-leg-length", or "shoulder-width". The dimensions and angles have to be the same in all views, but the pose can be different. Symmetry constraints, that the left and right body lengths are the same, are enforced as well. Minimizing only over angles, or only over model dimensions results in linear equations similar to what we have shown so far. Unfortunately the global minimization criteria over all parameters is a tri-linear equation system, that cannot be easily solved by simple matrix inversions. There are several possible techniques for minimizing such functions. We achieved good results with a Quasi-Newton method and a mixed quadratic and cubic line search procedure.

2.7 Model Fine Tuing

The supplied hand labels in the first frame should coincide with the location of the joints. We can roughly guess where the joints are, but still don't know exactly the underlying bone structure and their kinematics. We are able to extend the state space of our motion tracking framework, such that we also optimize over the kinematic model, and over the complete image sequence, instead of an image pair. Due to space limitations the details of the extended algorithm are described in the accompanying technical report [8].

3 Results

We applied this technique to video recordings in our lab and to photo-plate sequences of Eadweard Muybdrige's motion studies [21].

Figure 2: Example configurations of the estimated kinematic structure. First image shows the support maps of the initial configuration. In subsequent images the white lines show blob axes. The joint is the position on the intersection of two axes.

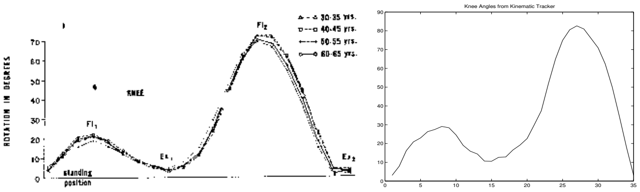

Figure 3: Comparison of a) data from [Murray et al] (left) and b) our motion tracker (right).

3.1 Single camera recordings

Our lab video recordings were done with a single camera. Therefore the 3D pose and some parts of the body can not be estimated completely. Figure 2 shows one example sequences of a person walking in a frontoparallel plane. We defined a 6 DOF kinematic structure: One blob for the body trunk, three blobs for the frontal leg and foot, connected with a hip joint, knee joint, and ankle joint, and two blobs for the arm connected with a shoulder and elbow joint. All joints have an axis orientation parallel to the Z-axis in the camera frame. The head blob was connected with one joint to the body trunk. The first image in figure 2 shows the initial blob support maps.

After the hand-initialization we applied the motion tracker to a sequence of 53 image frames. We could successfully track all body parts in this video sequence (see web-page). The video shows that the appearance of the upper leg changes significantly due to moving folds on the subject's jeans. The lower leg appearance does not change to the same extent. The constraints were able to enforce compatible motion vectors for the upper leg, based on more reliable measurements on the lower leg.

We can compare the estimated angular configurations with motion capture data reported in the literature. Murray, Brought, and Kory published [19] such measurements for the hip, knee, and angle joints. We compared our motion tracker measurements with the published curves and found good agreement. Figure 3.1a shows the curves for the knee and ankle reported in [19], and figure 3.1b shows our measurements.

We also experimented with a walking sequence of a subject seen from an oblique view with a similar kinematic model. As seen in figure 4, we tracked the angular configurations and the pose successfully over the complete sequence of 45 image frames. Because we use a scaled orthographic projection model, the perspective effects of the person walking closer to the camera had to be compensated by different scales. The tracking algorithm could successfully estimate

Figure 5: Eadweard Muybridge, The Human Figure in Motion, Plate 97: Woman Walking. The first 3 frames show part of a walk cycle from one example view, and the second 3 frames show the same time steps from a different view

the scale changes.

3.2 Digital Muybridge

The final set of experiments was done on historic footage recorded by Eadweard Muybridge in 1884 [21]. His methods are of independent interest, as they predate motion pictures. Muybridge had his models walk in an open shed. Parallel to the shed was a fixed battery of 24 cameras. Two portable batteries of 12 cameras each were positioned at both ends of the shed, either at an angle of 90 deg relative to the shed or an angle of 60 deg. Three photographs were take simultaneously, one from each battery. The effective 'framerate' of his technique is about two times lower then current video frame rates; a fact which makes tracking a harder problem.. It is to our advantage that he took for each time step three pictures from different viewpoints.

Figure 3.2 and figure 3.2 shows example photo plates. We could initialize the 3D pose by labeling all three views of the first frame and running the minimization procedure over the body dimensions and poses. Every body segment was visible in at least one of the three camera views, therefore we could track the left and the right side of the person. We applied this technique to a walking woman and a walking man. For the walking woman we had 10 time steps available that contained 60 % of a full walk cycle (figure 3.2). For this set of experiments we extended our kinematic model to 19 DOFs. The two hip joints, the two shoulder joints, and the neck joint, were modeled by 3 DOFs. The two knee joints and two elbow joints were modeled just by one rotation axis. Figure 3.2 shows the tracking results with the model overlayed. As you see, we could successfully track the complete sequence. To animate the tracking results we mirrored the left and right side angles to produce the remaining frames of a complete walk cycle. We animated the 3D motion capture data with a stick figure model and a volumetric model (figure 9), and it looks very natural. The video shows some of the tracking and animation sequences from several novel camera views, replicating the walk cycle performed over a century ago on the grounds of University of Pennsylvania.

For the visualization of the walking man sequence, we did not apply the mirroring, because he was carrying a boulder on his shoulder. This made the walk asymmetric. We reanimated the original tracked motion (figure 3.2) capture data for the man, and it also looked very natural.

4 Conclusion

In this paper, we have developed and demonstrated a new technique for articulated visual motion tracking. We

Figure 4: Example configurations of the estimated kinematic structure of a person seen from an oblique view.

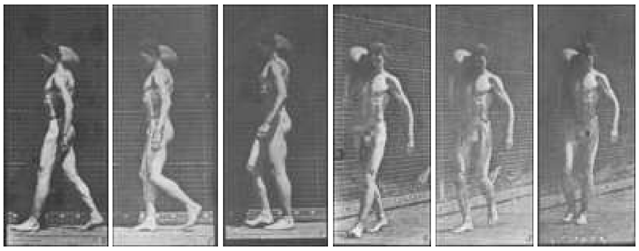

Figure 6: Eadweard Muybridge, The Human Figure in Motion, Plate 7: Man walking and carrying 75-LB boulder on shoulder. The first 3 frames show part of a walk cycle from one example view, and the second 3 frames show the same time steps from a different view

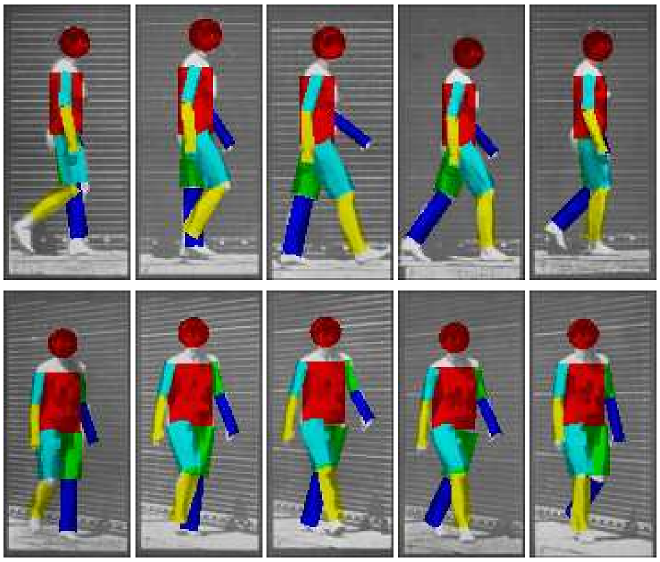

Figure 7: Muybridge's Woman Walking: Motion Capture results. This shows the tracked angular configurations and its volumetric model projected to 2 example views.

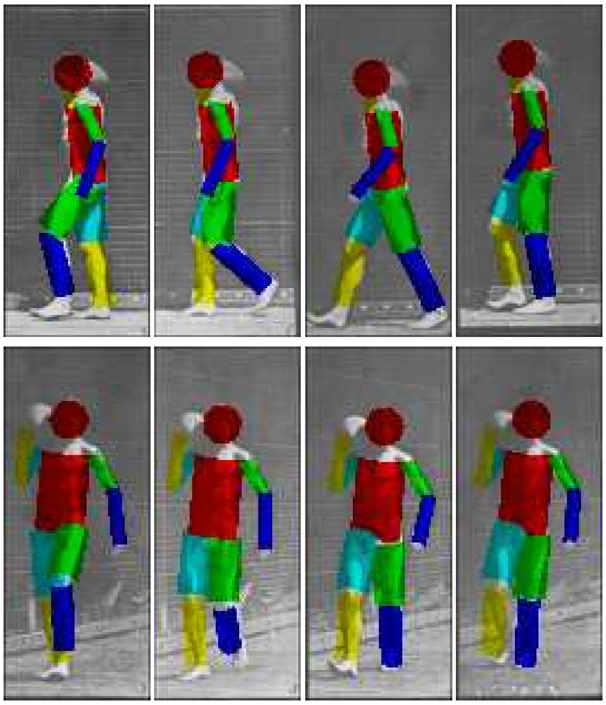

Figure 8: Muybridge's Man Walking: Motion Capture results. This shows the tracked angular configurations and its volumetric model projected to 2 example views.

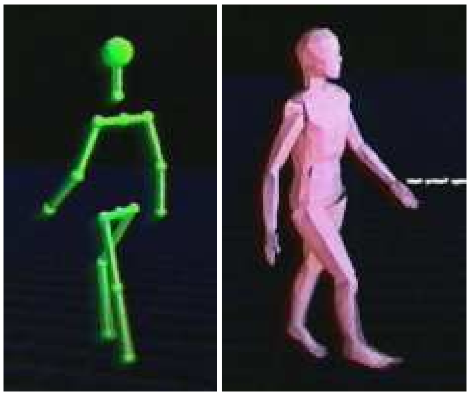

Figure 9: Computer models used for the animation of the Muybridge motion capture. Please check out the web-page to see the quality of the animation.

demonstrated results on video recordings of people walking both in frontoparallel and oblique views, as well as on the classic Muybridge photographic sequences recorded more than a century ago.

Visually tracking human motion at the level of individual joints is a very challenging problem. Our results are due, in large measure, to the introduction of a novel mathematical technique, the product of exponential maps and twist motions, and its integration into a differential motion estimation scheme. The advantage of this particular formulation is that it results in the equations that need to be solved to update the kinematic chain parameters from frame to frame being linear, and that it is not necessary to solve for any redundant or unnecessary variables.

Future work will concentrate on dealing with very large motions, as may happen, for instance, in videotapes of high speed running. The approach developed in this paper is a differential method, and therefore may be expected to fail when the motion from frame-to-frame is very large. We propose to augment the technique by the use of an initial coarse search stage. Given a close enough starting value, the differential method will converge correctly.

Acknowledgements

We would like to thank Charles Ying for creating the Open-GL animations, Shankar Sastry, Lara Crawford, Jerry Feldman, John Canny, and Jianbo Shi for fruitful discussions, Chad Carson for help in editing this document, and Interval Research Corp, and the California State MICRO program for supporting this research.

References

[1] Serge Ayer and Harpreet S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *Int. Conf. Computer Vision*, pages 777–784, Cambridge, MA., 1995.

[2] S. Basu, I.A. Essa, and A.P. Pentland. Motion regularization for model-based head tracking. In *International Conference on Pattern Recognition*, 1996.

[3] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, pages 237–252, 1992.

[4] M.J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, Jan 1996.

[5] M.J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *ICCV*, 1995.

[6] M.J. Black, Y.Yacoob, A.D.Jepson, and D.J.Fleet. Learning parameterized models of image motion. In *CVPR*, 1997.

[7] A. Blake, M. Isard, and D. Reynard. Learning to track the visual motion of contours. In *J. Artificial Intelligence*, 1995.

[8] C. Bregler and J. Malik. Video motion capture. Technical Report UCB//CSD-97-973, Comp. Sci. Dep., U.C. Berkeley, http://www.cs.berkeley.edu/ bregler/digmuy.html, 1997.

[9] E. Clergue, M. Goldber, N. Madrane, and B. Merialdo. Automatic face and gestual recognition for video indexing. In *Proc. of the Int. Workshop on Automatic Face- and Gesture-Recognition, Zurich, 1995*, 1995.

[10] L. Concalves, E.D. Bernardo, E. Ursella, and P. Perona. Monocular tracking of the human arm in 3d. In *Proc. Int. Conf. Computer Vision*, 1995.

[11] J.W. Davis and A.F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR*, 1997.

[12] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39, 1977.

[13] D.M. Gavrila and L.S. Davis. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *Proc. of the Int. Workshop on Automatic Face- and Gesture-Recognition, Zurich, 1995*, 1995.

[14] D. Hogg. A program to see a walking person. *Image Vision Computing*, 5(20), 1983.

[15] A. Jepson and M.J. Black. Mixture models for optical flow computation. In *Proc. IEEE Conf. Computer VIsion Pattern Recognition*, pages 760–761, New York, 1993.

[16] S.X. Ju, M.J.Black, and Y.Yacoob. Cardboard people: A parameterized model of articulated motion. In *2nd Int. Conf. on Automatic Face- and Gesture-Recognition, Killington, Vermont*, pages 38–44, 1996.

[17] I.A. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In *CVPR*, 1996.

[18] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. 7th Int. Joinnt Conf. on Art. Intell.*, 1981.

[19] M.P. Murray, A.B. Drought, and R.C. Kory. Walking patterns of normal men. *Journal of Bone and Joint Surgery*, 46-A(2):335–360, March 1964.

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

[21] Eadweard Muybridge. *The Human Figure In Motion*. Various Publishers, latest edition by Dover Publications, 1901.

[22] A. Pentland and B.Horowitz. Recovery of nonrigid motion and structure. *IEEE Transactions on PAMI*, 13(7):730–742, July 1991.

[23] J.M. Regh and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proc. Int. Conf. Computer Vision*, 1995.

[24] K. Rohr. Incremental recognition of pedestrians from image sequences. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 8–13, New York City, June, 1993.

[25] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[26] Yair Weiss and Edward H. Adelson. Perceptually organized EM: A framework for motion segmentaiton that combines information about form and motion. Technical Report 315, M.I.T Media Lab, 1995.

[27] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *SPIE Conference on Integration Issues in Large Commercial Media Delivery Systems*, volume 2615, 1995.