International Journal of Computer Vision 56(3), 179–194, 2004

Twist Based Acquisition and Tracking of Animal and Human Kinematics

CHRISTOPH BREGLER,*
Computer Science Department, Stanford University, Stanford, CA 94305, USA
chris.bregler@nyu.edu

JITENDRA MALIK
Computer Science Department, University of California at Berkeley, Berkeley, CA 94720, USA
malik@cs.berkeley.edu

KATHERINE PULLEN
Physics Department, Stanford University, Stanford, CA 94305, USA
pullen@graphics.stanford.edu

Received December 14, 1999; Revised May 27, 2003; Accepted May 30, 2003

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 and integration of a mathematical technique, the product of exponential maps and twist motions, 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. A new factorization technique lets us also recover the kinematic chain model itself. We are able to track several human walk cycles, several wallaby hop cycles, and two walk cycels of 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.

Keywords: human tracking, motion capture, kinematic chains, twists, exponential maps

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 (Bergen et al., 1992). These techniques rely on 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 (Jepson and Black, 1993; Ayer Sawhney, 1995; Weiss and Adelson, 1995). For each area, this approach still has the advantage that a large number of measurements are incorporated into

*Present address: Computer Science Dept., Courant Institute, Media Research Lab, 719 Broadway, 12th Floor, New York, NY 10003, USA. He was formerly at Stanford University.

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 (Murray et al., 1994). 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 (Hogg, 1983; Rohr, 1993; Regh and Kanade, 1995; Gavrila and Davis, 1995; Concalves et al., 1995; Clergue et al., 1995; Ju et al., 1996; Kakadiaris and Metaxas, 1996), 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 Wren et al. (1995) in tracking color blobs, and by Davis and Bobick (1997) in using motion templates. Nonrigid models were proposed by Pentland and Horowitz (1991), Blake et al. (1995), Black and Yacoob (1995) and Black et al. (1997).

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

The tracking technique of this paper has been presented in a shorter conference proceeding version in Bregler and Malik (1998). The new model acquisition technique has not been published previously.

2. Motion Estimation

We first describe a commonly used region-based motion estimation framework (Bergen and Anandan, 1992; Shi and Tomasi, 1994), and then describe the extension to kinematic chain constraints (Murray et al., 1994).

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:

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

$I(x, y, t)$ 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

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

The first-order Taylor series expansion of (1) leads to the commonly used gradient formulation (Lucas and Kanade, 1981):

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

$I_t(x, y)$ 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 Bergen et al., 1992 for details):

$$\mathbf{H} \cdot \phi + \vec{z} = \vec{0} \tag{4}$$

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

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

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 $w(x, y)$:

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

$\mathbf{W}$ 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 Murray et al. (1994)):

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

$q_o = [x_o, y_o, z_o, 1]^T$ 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:

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

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 Basu et al. (1996) propose a 3D ellipsoidal tracker based on Euler angles). In contrast, the *twist* representation provides a more elegant solution (Murray et al., 1994) 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:

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

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. It can be shown that for any arbitrary $\mathbf{G} \in SE(3)$ there exists a $\xi \in \Re^6$ twist representation. See (Murray et al., 1994) for more formal properties and a detailed geometric interpretation. It is convenient to drop the θ coefficient by relaxing the constraint that ω is unit length. Therefore $\xi \in \Re^6$.

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

$$\mathbf{G} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & d_x \\ r_{2,1} & r_{2,2} & r_{2,3} & d_y \\ r_{3,1} & r_{3,2} & r_{3,3} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \mathbf{e}^{\hat{\xi}} = \mathbf{I} + \hat{\xi} + \frac{(\hat{\xi})^2}{2!} + \frac{(\hat{\xi})^3}{3!} + \cdots \quad (10)$$

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:

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

s is the scale change of the scaled orthographic projection. The image motion of point $[x_{im}, y_{im}]$ from time t to time $t+1$ is:

$$\begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} = \begin{bmatrix} x_{im}(t+1) - x_{im}(t) \\ y_{im}(t+1) - y_{im}(t) \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$
$$\cdot \left(s(t+1) \cdot \mathbf{e}^{\hat{\xi}(t+1)} \cdot q_o - s(t) \cdot \mathbf{e}^{\hat{\xi}(t)} \cdot q_o \right)$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$
$$\cdot ((1 + \Delta s) \cdot \mathbf{e}^{\Delta\hat{\xi}} - \mathbf{I}) \cdot s(t) \cdot \mathbf{e}^{\hat{\xi}(t)} \cdot q_o$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot ((1 + \Delta s) \cdot \mathbf{e}^{\Delta\hat{\xi}} - \mathbf{I}) \cdot q_c \quad (12)$$

with

$$\mathbf{e}^{\hat{\xi}(t+1)} = \mathbf{e}^{\hat{\xi}(t)} \cdot \mathbf{e}^{\Delta\hat{\xi}}$$
$$s(t+1) = s(t) \cdot (1 + \Delta s) \quad (13)$$
$$q_c = s(t) \cdot \mathbf{e}^{\hat{\xi}(t)} \cdot q_o$$

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

$$(1 + \Delta s) \cdot \mathbf{e}^{\Delta\hat{\xi}} \approx (1 + \Delta s) \cdot \mathbf{I} + (1 + \Delta s) \cdot \Delta\hat{\xi} \quad (14)$$

and can rewrite (12) as:

$$\begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} = \begin{bmatrix} \Delta s & -\Delta\omega_z & \Delta\omega_y & \Delta v_1 \\ \Delta\omega_z & \Delta s & -\Delta\omega_x & \Delta v_2 \end{bmatrix} \cdot q_c \quad (15)$$

with

$$\Delta\xi = [\Delta v_1, \Delta v_2, \Delta v_3, \Delta\omega_x, \Delta\omega_y, \Delta\omega_z]^T$$

$\phi = [\Delta s, \Delta v_1, \Delta v_2, \Delta\omega_x, \Delta\omega_y, \Delta\omega_z]^T$ codes the relative scale and twist motion from time t to $t+1$. Note that (15) does not include Δv_3. Translation in the Z direction of the camera frame is not measurable under scaled orthographic projection.

2.2.3. 3D Geometric Model. Equation (15) describes the image motion of a point $[x_{im}, y_{im}]$ in terms of the motion parameters ϕ and the corresponding 3D point q_c in the camera frame. As previously defined in Eq. (7) q_c is a homogenous vector $[x, y, z, 1]^T$. It is the point that intersects the camera ray of the image point $[x_{im}, y_{im}]$ with the 3D model. The 3D model is given by the user (for example a cyclinder, superquadric, or polygonial model) or is estimated by an initialization procedure that we will describe below. The pose of the 3D model is defined by $G(t) = s(t) \cdot \mathbf{e}^{\hat{\xi}(t)}$. We assume $G(t)$ is the correct pose estimate for image frame $I(x, y, t)$ (the estimation result of this algorithm over the previous time frame). Since we assume scaled orthographic projection (11), $[x_{im}, y_{im}] = [x, y]$. We only need to determine z. In this paper we approximate the body segments by ellipsoidal 3D blobs. The 3D blobs are defined in the object frame. Following quadratic equation is the implicit function for the ellipsodial surface

with length $1/a_x$, $1/a_y$, $1/a_z$ along the x, y, z axis and centered around $M = [m_x, m_y, m_z, 1]^T$:

$$(q_o - M)^T \cdot \begin{bmatrix} a_x^2 & 0 & 0 & 0 \\ 0 & a_y^2 & 0 & 0 \\ 0 & 0 & a_z^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot (q_o - M) = 1 \quad (16)$$

Since $q_o = \mathbf{G}^{-1} q_c = \mathbf{G}^{-1}[x_{im}, y_{im}, z, 1]^T$ we can write the implicit function in the camera frame with:

$$\left(\mathbf{G}^{-1} \begin{bmatrix} x_{im} \\ y_{im} \\ z \\ 1 \end{bmatrix} - M \right)^T \cdot \begin{bmatrix} a_x^2 & 0 & 0 & 0 \\ 0 & a_y^2 & 0 & 0 \\ 0 & 0 & a_z^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \left(\mathbf{G}^{-1} \begin{bmatrix} x_{im} \\ y_{im} \\ z \\ 1 \end{bmatrix} - M \right) = 1 \quad (17)$$

Therefore z is the solution of this quadratic Eq. (17). For image points that are inside the blob it has 2 (close-form) solutions. We pick the smaller solution (z value that is closer to the camera). Using (17) we can calculate for all points inside the blob the q_c points. For points outside the blob it has no solution. Those points will not be part of the estimation setup.

For more complex 3D shape models, the z calculation can be replaced by standard graphics ray-casting algorithms. We have not implemented this generalization yet.

2.2.4. Combining 3D Motion and Geometric Model. Inserting (15) into (3) leads to following equation for each point $[x_i, y_i]$ inside the blob:

$$\begin{aligned} & I_t + I_x \cdot [\Delta s, -\Delta\omega_z, \Delta\omega_y, \Delta v_1] \cdot q_c \\ & \quad + I_y \cdot [\Delta\omega_z, \Delta s, -\Delta\omega_x, \Delta v_2] \cdot q_c = 0 \\ & \Leftrightarrow I_t(i) + H_i \cdot [s, \Delta v_1, \Delta v_2, \Delta\omega_x, \Delta\omega_y, \Delta\omega_z]^T = 0 \end{aligned} \quad (18)$$

$$\begin{aligned} H_i = [& I_x \cdot x_i + I_y \cdot y_i, I_x, I_y, -I_y \cdot z_i, I_x \cdot z_i, \\ & - I_x \cdot y_i + I_y \cdot x_i] \in \Re^{1\times 6} \quad \text{with} \\ I_t := I_t(x_i, y_i), & \; I_x := I_x(x_i, y_i), \; I_y := I_y(x_i, y_i) \end{aligned}$$

For N pixel positions we have N equations of the form (18). This can be written in matrix form:

$$\mathbf{H} \cdot \phi + \vec{z} = \mathbf{0} \quad (19)$$

with

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

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

2.2.5. 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 (Fig. 1). This is a revolute joint, and can be modeled by a twist (Murray et al.,

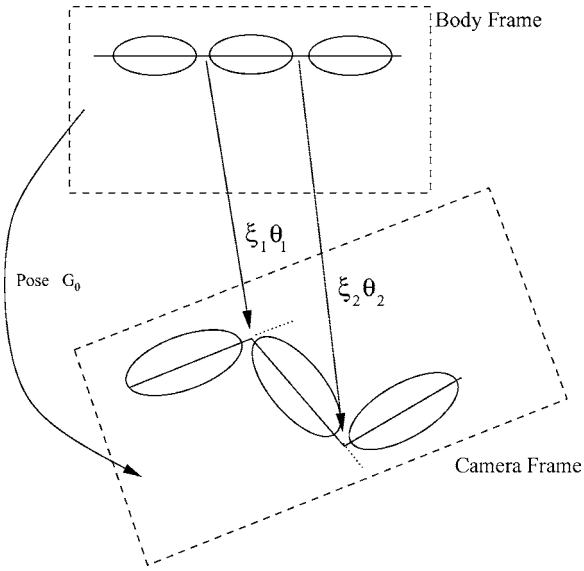


Figure 1. Kinematic chain defined by twists.

1994):

$$\xi_1 = \begin{bmatrix} -\omega_1 \times q_1 \\ \omega_1 \end{bmatrix} \tag{20}$$

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

$$\mathbf{g_1} = e^{\hat{\xi_1} \cdot \theta_1} \tag{21}$$

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

$$\begin{aligned} \mathbf{g}(\theta_1) &= \mathbf{G_0} \cdot e^{\hat{\xi_1} \cdot \theta_1} \\ q_c &= \mathbf{g}(\theta_1) \cdot q_o \end{aligned} \tag{22}$$

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 $\theta_1, \theta_2, \ldots, \theta_k$:

$$\begin{aligned} &\mathbf{g_k}(\theta_1, \theta_2, \ldots, \theta_k) \\ &\quad = \mathbf{G_0} \cdot e^{\hat{\xi_1} \cdot \theta_1} \cdot e^{\hat{\xi_2} \cdot \theta_2} \cdot \cdots \cdot e^{\hat{\xi_k} \cdot \theta_k} \end{aligned} \tag{23}$$

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 $\xi_1', \xi_2', \ldots, \xi_k'$ and the angular velocities $\dot{\theta}_1, \dot{\theta}_2, \ldots, \dot{\theta}_k$:

$$V_k = \xi_1' \cdot \dot{\theta}_1 + \xi_2' \cdot \dot{\theta}_2 + \cdots \xi_k' \cdot \dot{\theta}_k \tag{24}$$

The twists ξ_k' are coordinate transformations of ξ_k. The coordinate transformation for ξ_k' is done relative to $\mathbf{g_{k-1}}$ (as defined in (23)) and can be computed with a so called *Adjoint* transformation $\mathbf{Ad_{g_{k-1}}}$ (Murray et al., 1994). If R is the rotation matrix of g_{k-1} and $\hat{p}$ is the translation vector of g_{k-1} ($g_{k-1} = \left[\begin{smallmatrix} R & \hat{p} \\ \mathbf{0} & 1 \end{smallmatrix}\right]$) then we can calculate a 6×6 adjoint matrix:

$$\mathbf{Ad}_{g_{k-1}} = \begin{bmatrix} R & \hat{p} \cdot R \\ \mathbf{0} & R \end{bmatrix} \tag{25}$$

ξ_k' is computed in multiplying the adjoint matrix to ξ_k:

$$\xi_k' = \mathbf{Ad}_{g_{k-1}} \xi_k \tag{26}$$

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:

$$\begin{aligned} \begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ &\quad \cdot [\hat{\xi}_1' \cdot \dot{\theta}_1 + \hat{\xi}_2' \cdot \dot{\theta}_2 + \cdots + \hat{\xi}_k' \cdot \dot{\theta}_k] \cdot q_c \end{aligned} \tag{27}$$

Recall (18) relates the image motion of a point q_c to changes in pose $\mathbf{G}_0$. We combine (18) and (27) to relate the image motion to the combined vector of pose change and angular change $\Phi = [\Delta s, v_1', v_2', \Delta\omega_x, \Delta\omega_y, \Delta\omega_z, \dot{\phi}_1, \dot{\phi}_2, \ldots, \dot{\phi}_K]^T$:

$$\begin{aligned} &I_t + H_i \cdot [s, v_1', v_2', \Delta\omega_x, \Delta\omega_y, \Delta\omega_z]^T \\ &\quad + J_i \cdot [\dot{\theta}_1, \dot{\theta}_2, \ldots \dot{\theta}_K]^T = 0 \end{aligned} \tag{28}$$

$$[\mathbf{H}, \mathbf{J}] \cdot \Phi + \vec{z} = \mathbf{0} \tag{29}$$

with

$$\begin{aligned} \mathbf{J} &= \begin{bmatrix} J_1 \\ J_2 \\ \ldots \\ J_N \end{bmatrix} \quad \text{and } \mathbf{H}, \vec{z} \text{ as before} \\ J_i &= [J_{i,1}, J_{i,2}, \ldots, J_{i,K}] \\ J_{i,k} &= \begin{cases} [I_x, I_y] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \hat{\xi}_k \cdot q_c \\ 0 \quad \text{if pixel } i \text{ is on a segment that is not affected by joint } \xi_k \end{cases} \end{aligned} \tag{30}$$

The least squares solution to (29) is:

$$\Phi = -([\mathbf{H}, \mathbf{J}]^T \cdot [\mathbf{H}, \mathbf{J}])^{-1} \cdot [\mathbf{H}, \mathbf{J}]^T \cdot \vec{z} \tag{31}$$

Φ 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 Φ. Based on the re-warped image we compute the new image gradients. Repeating this process of warping and solving (31) 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 (29)):

$$[\mathbf{H_c}, \mathbf{J_c}] \cdot \begin{bmatrix} \Omega_c \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dots \\ \dot{\phi}_K \end{bmatrix} + \vec{z}_c = \mathbf{0} \tag{32}$$

$\Omega_c = [s'_c, v'_{1,c}, v'_{2,c}, \omega'_{x,c}, \omega'_{y,c}, \omega'_{z,c}]^T$ describes the pose seen from view c. All views share the same angular parameters, because the cameras are triggered at the same time. We can simply combine all C equation systems into one large equation system:

$$\begin{bmatrix} \mathbf{H_1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{J_1} \\ \mathbf{0} & \mathbf{H_2} & \dots & \mathbf{0} & \mathbf{J_2} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{H_C} & \mathbf{J_C} \end{bmatrix} \cdot \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \dots \\ \Omega_C \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dots \\ \dot{\phi}_K \end{bmatrix} + \begin{bmatrix} \vec{z}_1 \\ \vec{z}_2 \\ \dots \\ \vec{z}_C \end{bmatrix} = \mathbf{0} \tag{33}$$

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 least squares estimation (31) can be generalized to a weighted least squares estimation:

$$\Phi = -((\mathbf{W}_k \cdot [\mathbf{H}, \mathbf{J}])^T \cdot [\mathbf{H}, \mathbf{J}])^{-1} \cdot (\mathbf{W}_k \cdot [\mathbf{H}, \mathbf{J}])^T \vec{z} \tag{34}$$

$\mathbf{W_k}$ is a diagonal matrix that codes the support map for segment k. The values along the diagonal of the matrix are the different weights for each pixel location. If we only allow values 0 and 1 for the weights, we do exactly the same as in (30). If the value is 1, that specific pixel is used in the estimation (that specific row in $[\mathbf{H}, \mathbf{J}]$ is multiplied by 1). If the value is 0, that specific pixel is discarded in the estimation (that specific row in $[\mathbf{H}, \mathbf{J}]$ is multiplied by 0). With continuous weight values between 0 and 1 the different pixels (rows in $[\mathbf{H}, \mathbf{J}]$) contribute with different strength to the final solution.

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. Sometimes a few outliers (fast motion from the background or other errors) can dominate the estimatimation and cause larger erros. Robust statistics would be one solution to this problem (Black and Anandan, 1996). Another solution is an EM-based layered representation (Ayer and Sawhney, 1995; Dempster et al., 1977; Jepson and Black, 1993; Weiss and Adelson, 1996) that compute for those pixel locations low weight values.

We use the EM-based solution for fine tuning the shape of the support maps $\mathbf{W_k}$. EM (Expectation Maximization) is a iterative maximum-likelihood estimation technique. Work by Ayer and Sawhney (1995), Jepson and Black (1993) and Weiss and Adelson (1996) proposed to use this technique to iteratively estimate motion models and support maps.

We start with an initial guess of the support map (all weights inside the ellipsoidal projection are set to 1). Given the initial $\mathbf{W}_k$, we iterate between the M-step and E-steps. The M-step is the application of Eq. (34) to all body segments. The result are new twist motions Φ for all segments. Using those parameters, we can calculate the posteriori probabilities for each pixel location that it belongs to the specific segment k. It is done in the same way as in Ayer and Sawhney (1995): For each pixel location i the difference d_i of current frame t warped by the estimated motion and the next frame at $t + 1$ is computed. Assuming a zero mean gaussian noise model of the pixel difference, the posteriory probabilites for each pixel i are computed and assigned to W_k. For the results reported in this paper we only iterate once.

2.5. Tracking Recipe

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

- **Input**: $I(t), I(t+1), \mathbf{G_0(t)}, \theta_1(t), \theta_2(t), \ldots, \theta_K(t)$ (Two images and the pose and angles for the first image).
- **Output:** $\mathbf{G_0(t+1)}, \theta_1(t+1), \theta_2(t+1), \ldots, \theta_K(t+1)$. (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 (34)
   (c) Update $G_0(t+1) := G_0(t+1) \cdot (1+\Delta s) \cdot e^{\frac{\hat{\xi}'}{1+\Delta s}}$
   (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 Tuning (Factorization Based Kinematic Model Reconstruction)

The above method assumes that we have a correct model for the locations of the joints. However, in reality, it is often difficult to measure the exact joint positions, which may in turn affect the accuracy of the method. If we extend the state space of our motion tracking framework to include a sequence of more than two images, we are able to iteratively solve for the joint locations, and thus determine the kinematic model directly from the video data.

Our technique starts with an initial guess of the kinematic model $\xi_1, \ldots, \xi_k$. Given the initial guess, we compute for each time frame t the pose $\xi_0(t)$, and all angles $\theta_1(t), \ldots, \theta_k(t)$ (using the tracking technique described in the previous sections). Given all poses and angles, we can recompute a better fitting kinematic model, and re-iterate.

We can rewrite (29), such that it is parameterized by by a specific twist ξ_l:

$$I_t + H_i \cdot [s, v_1', v_2', \Delta\omega_x, \Delta\omega_y, \Delta\omega_z]^T + J_i \cdot [\dot{\theta}_1, \dot{\theta}_2, \ldots \dot{\theta}_K]^T = 0 \quad (35)$$

$$C_i + J_i \cdot [\dot{\theta}_1, \dot{\theta}_2, \ldots \dot{\theta}_K]^T = 0 \quad (36)$$

$$C_i + \left(\sum_{k \neq l} J_{i,k} \cdot \dot{\theta}_k \right) + J_{i,l} \cdot \dot{\theta}_l = 0 \quad (37)$$

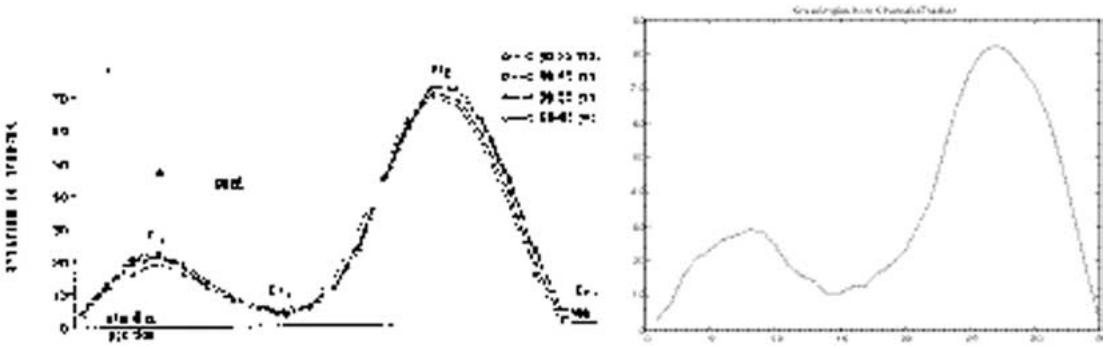
$$D_i + J_{i,l} \cdot \dot{\theta}_l = 0 \quad (38)$$

$$D_i + [I_x, I_y, 0, -I_y \cdot z, I_x \cdot z, -I_x \cdot y + I_y \cdot x] \cdot \mathbf{Ad}_{\mathbf{g_{l-1}}} \cdot \xi_l \cdot \dot{\theta}_l = 0 \quad (39)$$

$$D_i + M_i \cdot \xi_l \cdot \dot{\theta}_l = 0 \quad (40)$$

The scalar D_i and the 1×6 vector M_i contain all the spatio-temporal gradients I_x, I_y, I_t and 3D point locations x, y, z for image point at location i. Stacking all N equations together for all N pixel locations leads

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.



(a) (b)

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

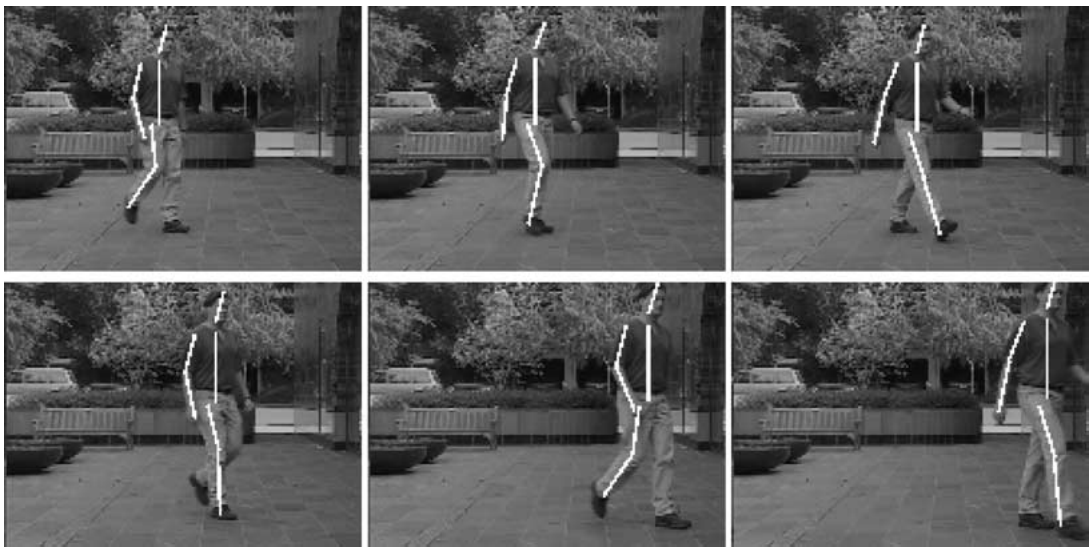


Figure 4. Example configurations of the estimated kinematic structure of a person seen from an oblique view.

to another system of equations:

$$M \cdot \xi_l \cdot \dot{\theta}_l = D$$

$$\text{with} \quad M = \begin{bmatrix} M_1 \\ M_2 \\ \dots \\ M_N \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} -D_1 \\ -D_2 \\ \dots \\ -D_N \end{bmatrix} \tag{41}$$

We can write out the least square solution:

$$\xi_l \cdot \dot{\theta}_l = (M^T \cdot M)^{-1} \cdot D = E \tag{42}$$

Equation (42) descibes only one specific instance in time. Computing E for all time steps let us write following bilinear equation:

$$\xi_l \cdot [\dot{\theta}_l(1), \dot{\theta}_l(2), \dots, \dot{\theta}_l(T)] = [E(1), E(2), \dots, E(T)] \tag{43}$$

$$\xi_l \cdot [\dot{\theta}_l(1), \dot{\theta}_l(2), \dots \dot{\theta}_l(T)] = W \tag{44}$$

The right side contains a $6 \times T$ matrix W. As derived above, W is computed from all spatio-temporal gradient measurements at all pixels and all time instances, and from the current guess of the kinematic model and angles.

The left side is the twist ξ_l multiplied with all angular velocities over the entire time period. The structure of this equation tells us, that W is of rank 1. Similar to the Tomasi-Kanade factorization (Tomasi and Kanade, 1992) of a tracking matrix into a pose and shape matrix, we can factor W into a twist and angular velocity matrix. Using SVD, ξ_l is a normal vector. The constraint that only the lower part of the twist $(\omega_x, \omega_y, \omega_z)$ has to be normal can be enforced with a simple rescaling of the SVD solution.

Our reconstruction algorithm computes this factorization for each twist ξ_l. Given the new more accurate twist model, it re-tracks the entire footage to compute new poses and angles. It then iterates.

3. Results

We applied this technique to video recordings in our lab, to photo-plate sequences of Eadweard

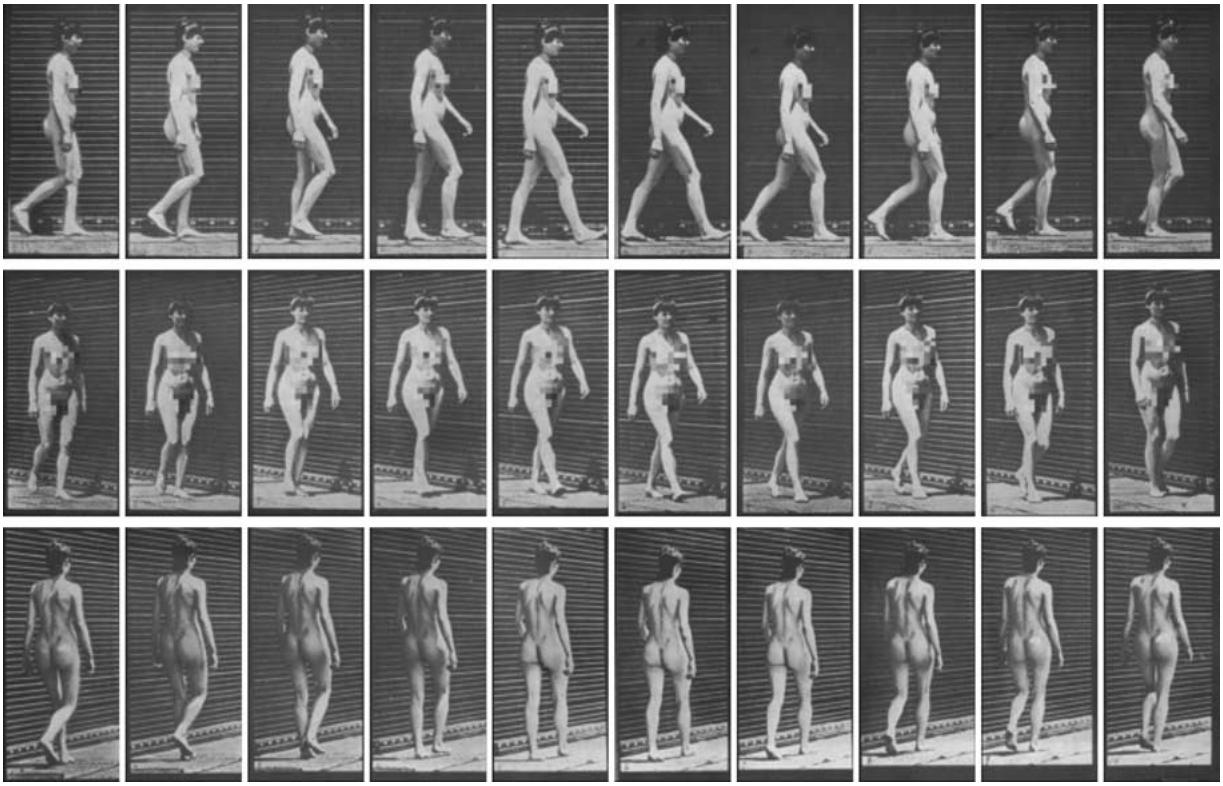


Figure 5. Eadweard Muybridge, the human figure in motion, Plate 97: Woman walking. The first row show a walk cycle from one example view, and the second and third row shows the same time steps from a different views.

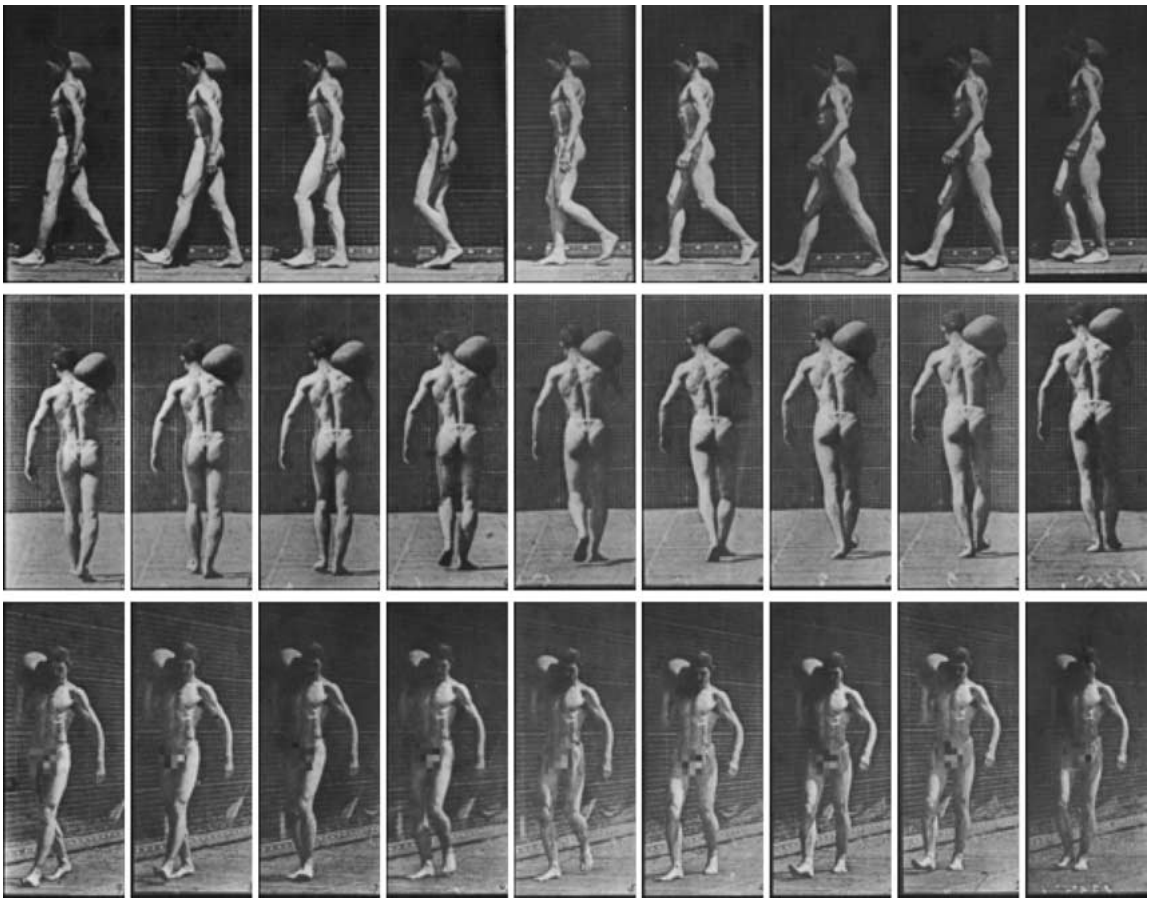


Figure 6. Eadweard Muybridge, The human figure in motion, Plate 7: Man walking and carrying 75-LB boulder on shoulder. The first row shows part a walk cycle from one example view, and the second and third row shows the same time steps from different views.

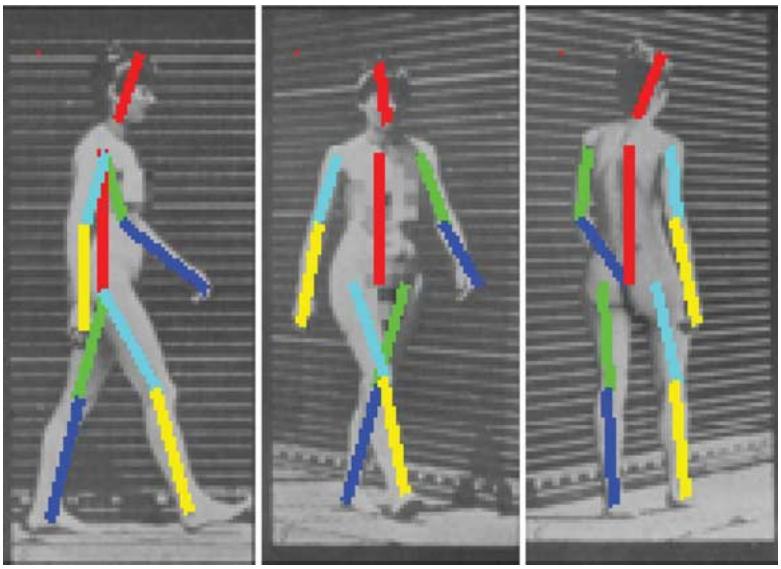


Figure 7. Initialization of Muybridge's woman walking: This visualizes the initial angular configuration projected to 3 example views.

Muybdrige's motion studies (Muybridge, 1901), and to Wallaby Hopping sequences

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 Fig. 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 (Murray et al., 1964) 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(a) shows the curves for the knee and ankle reported in Murray et al. (1964) and Fig. 3(b) 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 Fig. 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

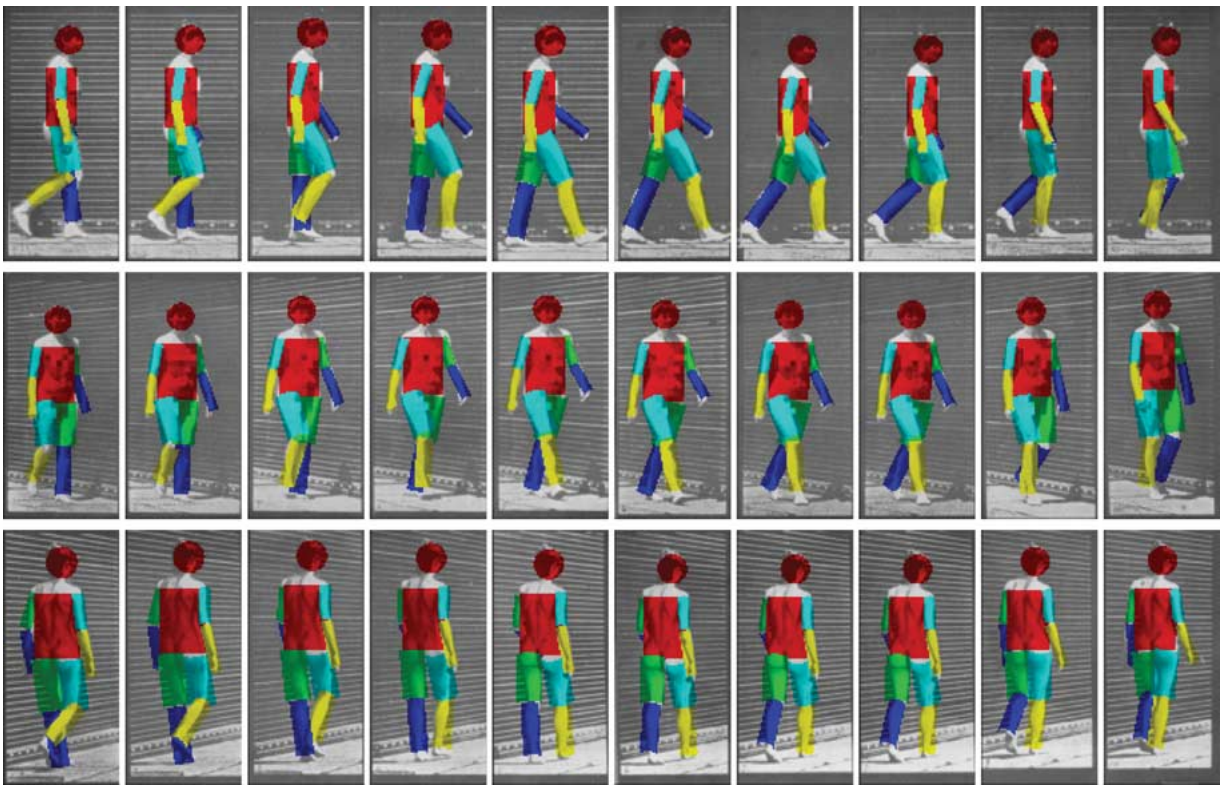


Figure 8. Muybridge's woman walking: Motion capture results. This shows the tracked angular configurations and its volumetric model projected to all 3 example views.

effects of the person walking closer to the camera had to be compensated by different scales. The tracking algorithm could successfully estimate the scale changes.

3.2. *Digital Muybridge*

The next set of experiments was done on historic footage recorded by Eadweard Muybridge in 1884 (Muybridge, 1901). 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.

Figures 5 and 6 shows example photo plates. We initialize the 3D pose by labeling all three views of the first frame and running the minimization procedure over the body dimensions and poses. Figure 7 shows one example initialization. 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 (Fig. 5). For this set of experiments we extended

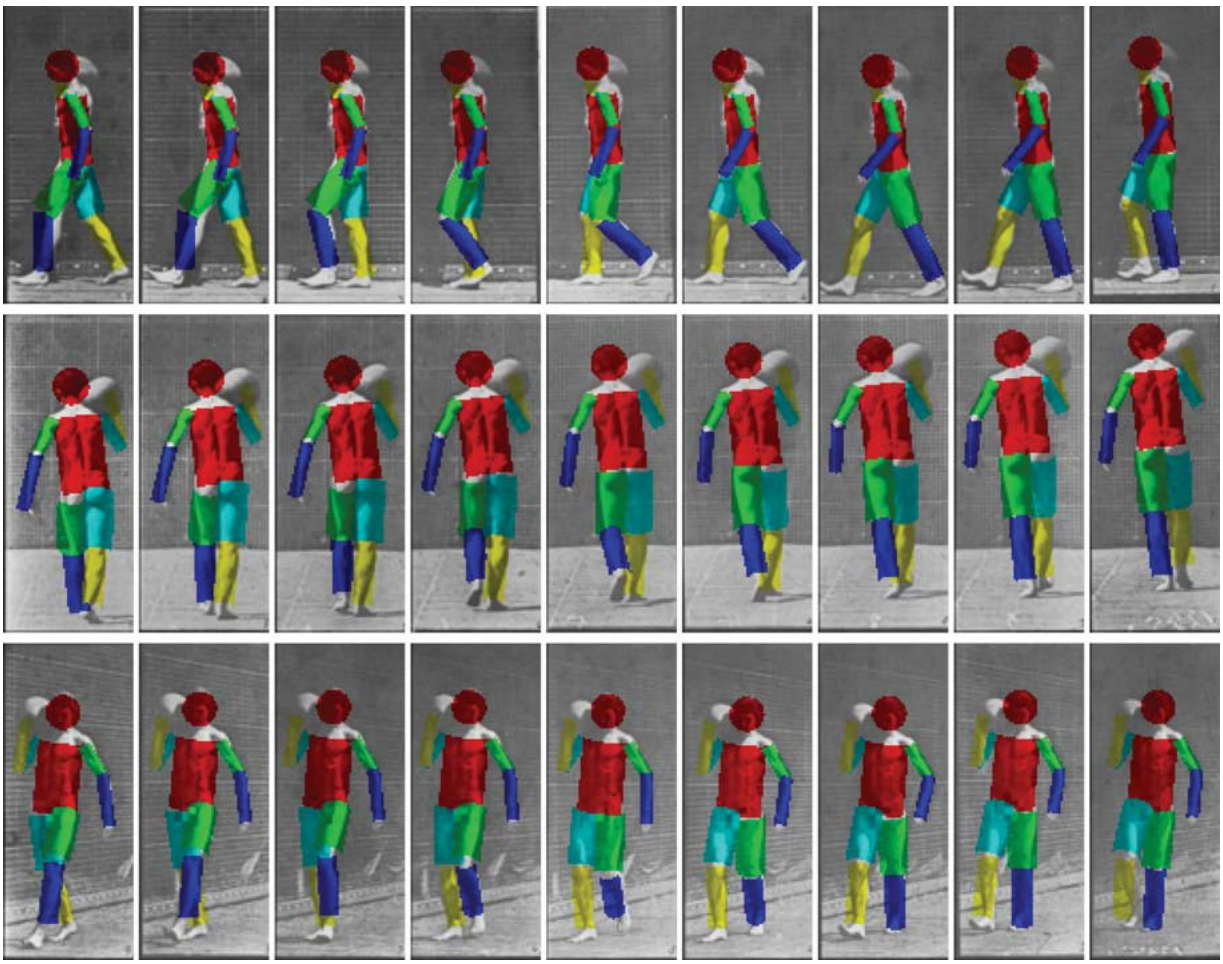


Figure 9. Muybridge's man walking: Motion capture results. This shows the tracked angular configurations and its volumetric model projected to all 3 example views.

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 8 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(Fig. 10), 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 re-animated the original tracked motion (Fig. 9) capture data for the man, and it also looked very natural.

3.3. Acquistion of Kinematic Models for Wallaby Recordings

As an initial test of the fitting technique described in Section 2.7, we used video data of a wallaby (a small species of kangaroo) hopping on a treadmill. The animal had markers placed on its joints, as the data was originally intended for biomechanical studies of the forces on its joints. However, it was clear that measuring the locations of the markers and computing the angles directly from that data would not be accurate,

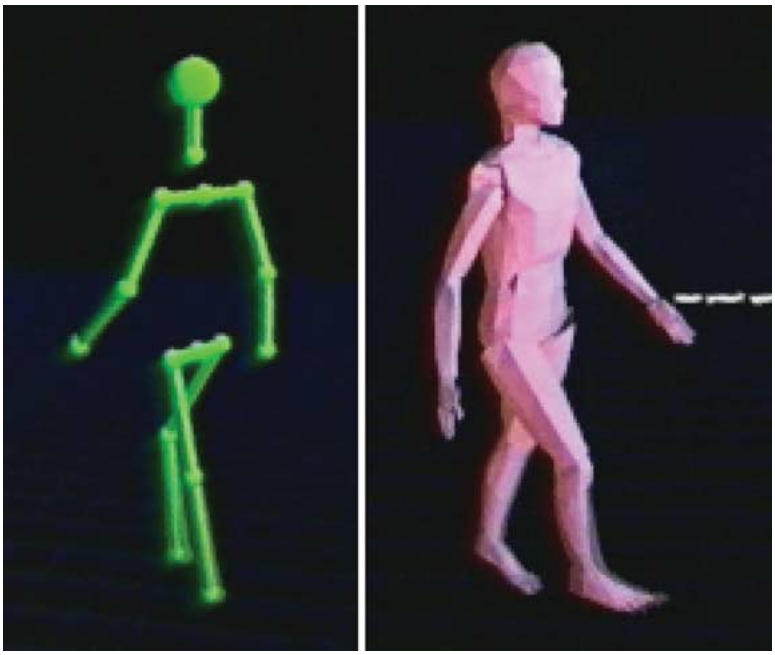


Figure 10. Computer models used for the animation of the Muybridge motion capture. Please check out the web-page to see the quality of the animation.

Figure 11. Hopping Wallaby and acuired kinematic model overlayed.

as the distance between any given pair of consecutive markers (for example, the hip and knee markers) varied by up to 50% over one hop cycle due to the soft deformations of the skin and muscle. As a result, this is a situation where a method such as ours that could actually determine the kinetmatic structure of the animal would be valuable.

Equation (44) is greatly simplified in 2D, because ω_x and ω_y are zero. Because the wallaby hops with its legs together, it is a valid approximation to assume the motion occurs in a plane. The frame rate of the data was 250 fps, yielding roughly 80 frames per hop cycle. As an initial guess for the kinematic model at each time, the markers on the joints were used. Then 8–10 succesive frames were used to solve for the twist parameters. When this process was repeated over a series of initial time points, we achieved consistant results for the limb lengths. Results are shown in Fig. 11 in which we have overlayed the resulting model on the images.

4. Conclusion

In this paper, we have developed and demonstrated a new technique for articulated visual motion tracking and acquisition. We demonstrated results on video recordings of animals and people hopping and walking both in frontoparallel and oblique views, as well as on the classic Muybridge photographic sequences recorded more than a century ago.

Visually tracking and acquistion of animal and 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.

Acknowledgments

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, Ana Rabinowicz for providing the walaby data, and Interval Research Corp, the California State MICRO program and the Nation Science Foundation for supporting this research.

References

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

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

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

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

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

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

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

Bregler, C. and Malik, J. 1998. Estimating and tracking kinematic chains. In *IEEE Conf. On Computer Vision and Pattern Recognition.*

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

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

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

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

Gavrila, D.M. and Davis, L.S. 1950. 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.

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

Jepson, A. and Black, M.J. 1993. Mixture models for optical flow computation. In *Proc. IEEE Conf. Computer Vision Plattern Recognition*, New York, pp. 760–761.

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

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

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

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

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

Muybridge, E. 1901. *The Human Figure in Motion*. Various Publishers, latest edition by Dover Publications.

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

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

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

Shi, J. and Tomasi, C. 1994. Good features to tract. In *CVPR*.

Tomasi, C. and Kanade, T. 1992. Shape and motion from image streams under orthography: A factorization method. *Int. J. of Computer Vision*, 9(2):137–154.

Weiss, Y. and Adelson, H.E. 1995. Perceptually organized EM: A framework for motion segmentation that combines information about form and motion. Technical Report 315, M.I.T Media Lab.

Weiss, Y. and Adelson, H.E. 1996. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Proc. IEEE Conf. Computer Vision Pattern Recognition*.

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