James M. Rehg

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332
rehg@cc.gatech.edu

Daniel D. Morris
Takeo Kanade

Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213
ddmorris@ri.cmu.edu
tk@ri.cmu.edu

Ambiguities in Visual Tracking of Articulated Objects Using Two- and Three-Dimensional Models

Abstract

Three-dimensional (3D) kinematic models are widely-used in video-based figure tracking. We show that these models can suffer from singularities when motion is directed along the viewing axis of a single camera. The single camera case is important because it arises in many interesting applications, such as motion capture from movie footage, video surveillance, and vision-based user-interfaces.

We describe a novel two-dimensional scaled prismatic model (SPM) for figure registration. In contrast to 3D kinematic models, the SPM has fewer singularity problems and does not require detailed knowledge of the 3D kinematics. We fully characterize the singularities in the SPM and demonstrate tracking through singularities using synthetic and real examples.

We demonstrate the application of our model to motion capture from movies. Fred Astaire is tracked in a clip from the film "Shall We Dance". We also present the use of monocular hand tracking in a 3D user-interface. These results demonstrate the benefits of the SPM in tracking with a single source of video.

KEY WORDS—**AUTHOR: PLEASE PROVIDE**

1. Introduction

The kinematics of an articulated object provide the most fundamental constraint on its motion, and there has been a significant amount of research into the use of 3D kinematic models for visual tracking of humans (Sidenbladh, Black, and Fleet 2000; Deutscher, Blake, and Reid 2000; DiFranco, Cham, and Rehg 2001; Bregler and Malik 1998; Kakadiaris and Metaxas 1996; Gavrila and Davis 1996; Rehg 1995; Yamamoto and Koshikawa 1991; Hogg 1983). Kinematic models play two roles in tracking. First, they define the desired output—a state vector of joint angles that encodes the three-dimensional (3D) configuration of the model. Secondly, they define the mapping between states and image measurements that makes registration possible.

In situations where multiple camera views are available for tracking an articulated object, the 3D kinematic constraint is particularly powerful. With enough cameras, the state space is fully observable, meaning that the available image measurements are sufficient to determine an estimate for each state in the model. As uncertainty in the 3D state estimate is reduced, the 3D kinematic model provides an increasingly powerful constraint on the image motion. Unfortunately in certain tracking applications, such as 3D motion capture from movie footage (Rehg 2000) or visual surveillance, only a single (monocular) video source is available. The result is a loss of observability and an increase in the uncertainty of the 3D state (Shimada et al. 1998; Rehg 1995).

In many cases, the constraint on image motion due to the 3D kinematics is expressed as the visual kinematic Jacobian, a linearized map from joint angles to image measurements. The Jacobian is a key element in nonlinear least-squares tracking algorithms (Kakadiaris and Metaxas 1996; Rehg and Kanade 1994b; Pentland and Horowitz 1991) as well as in techniques for "accelerating" the convergence of stochastic tracking algorithms based on particles (Heap and Hogg 1998) or kernels (Cham and Rehg 1999). The singularities of this Jacobian correspond to state space regions in which there is a local loss of observability. Kinematic ambiguity in reconstructing the 3D motion is compounded by other sources of ambiguity including self-occlusions and background clutter.

The International Journal of Robotics Research
Vol. 22, No. 6, June 2003, pp. xxx-xxx,
©2003 Sage Publications

Recent approaches to monocular figure tracking have addressed these challenges through a variety of innovations, including learned motion models (Sidenbladh, Black, and Fleet 2000; Howe, Leventon, and Freeman 1999; Brand 1999; Pavlović et al. 1999), efficient stochastic estimators for large state spaces (Deutscher, Blake, and Reid 2000; MacCormick and Isard 2000; Heap and Hogg 1998), and appearance manifolds (Toyama and Blake 2001). These algorithms can be seen as defining the state-of-the-art for visual tracking in general, as a result of their sophistication and their performance on challenging problems.

In contrast to this recent progress, however, there has been relatively little work done on analyzing systematically the sources of error in articulated tracking, as well as the optimality of the many different components which make up a successful tracking system. As a consequence, it is often unclear what the best solution approach is for a given articulated tracking problem. This paper represents an initial step in addressing this deficiency. We analyze the effectiveness and limitations of kinematic models in both two-dimensional (2D) and 3D approaches to figure tracking.

This paper makes three main contributions. First, we characterize the ambiguities that result in tracking with both 2D and 3D kinematic models. This characterization includes an analysis of the singularites of the visual kinematic Jacobian in both cases. Secondly, we propose a novel scaled prismatic model (SPM) for 2D figure registration. This model is derived by projecting a 3D kinematic model into the image plane. Thirdly, we present experimental results for tracking both figures and hands in 2D and 3D using a common framework for the kinematics and appearance modeling. Our results include a comparison between 2D and 3D models on real data. Preliminary versions of this work appeared in Morris and Rehg (1998), Rehg (1995) and Rehg and Kanade (1994a).

2. Appearance Models for 3D Articulated Object Tracking

A key step in any tracking approach is to specify a likelihood model $p(\mathbf{z}|\mathbf{x})$ which describes the appearance of the tracked object in the image $\mathbf{z}$ conditioned on the state vector $\mathbf{x}$. In probabilistic tracking algorithms, this model is combined with a prior model $p(\mathbf{x})$ using Bayes rule. Examples of this approach include the extended Kalman filter and various particle filters. In deterministic tracking algorithms, the likelihood is treated as an objective function which is minimized numerically.[1]

In both deterministic and probabilistic tracking algorithms, it is often useful to evaluate the gradient of the likelihood to accelerate the convergence of the tracker. This is the standard approach in quasi-Newton and Newton methods for deterministic tracking. It is also a common method of "accelerating" particle filter and kernel-based trackers (Cham and Rehg 1999; Heap and Hogg 1998). The gradient computation is expressed as a product of Jacobians which collectively map state space velocities into the velocities of image features.

In this section we will analyze the structure of the likelihood function and its gradient when $\mathbf{x}$ is the state vector for a 3D kinematic model and $\mathbf{z}$ is modeled by template features. This analysis has relevance to both probabilistic and deterministic tracking algorithms. We will show that the structure of the Jacobian is particularly useful in identifying ambigous or ill-posed conditions for 3D tracking.

2.1. Template Plane Model of Link Appearance

In classical robotic kinematic analysis, the shape of each link in a manipulator is abstracted away by the joint transform. In kinematic analysis for visual tracking, however, the shape of each link is intimately tied to its visual appearance. Therefore, one of the key issues in any articulated tracking system is the specification of the link appearance model. Various approaches are surveyed in Gavrila and Davis (1996).

Highly articulated objects such as the human body or a robot manipulator tend to have links with simple shapes. For example, finger phalanges and the limbs of the body are approximately cylindrical. Even when there are significant asymmetries due to clothing or soft tissue deformations, these links typically have a central axis which is defined by a pair of joint centers. Figure 1 illustrates the basic geometry. Link i is attached to the kinematic chain via joints i and $i+1$. The vector $\hat{\mathbf{a}}_i$ which connects the two joint centers defines the central axis for the link.[2] For terminal links, $\hat{\mathbf{a}}_i$ can be defined arbitrarily, and is usually set to the approximate axis of symmetry for the link shape.

Many 3D articulated tracking systems have employed cylinders, ellipsoids, or superquadrics to model the shape of the links. In some cases, the occluding limbs of the shape models are matched to image curves during tracking. In other approaches, the link shapes are used to define an image-flow or texture-based appearance model. In this section we will describe and analyze the template plane appearance model introduced in Rehg (1995) and Rehg and Kanade (1995). It is a view-based representation of the link appearance with respect to its central axis. It is arguably the simplest useful appearance model for articulated tracking.

In the template plane model, the image appearance of the kth link in a kinematic chain is modeled by the projection of a view-dependent texture-mapped plane, which is embedded in the joint coordinate frame for the link. The template plane passes through the joint centers for the link and is oriented towards the camera. "Pasted" onto the plane is a template of

1. While deterministic trackers are limited by their inability to handle occlusions and background clutter, they can still be useful in certain applications where the background can be controlled. We will discuss one example in Section 6.1.

2. We will write $\hat{\mathbf{x}}$ when $\mathbf{x}$ is a unit vector.

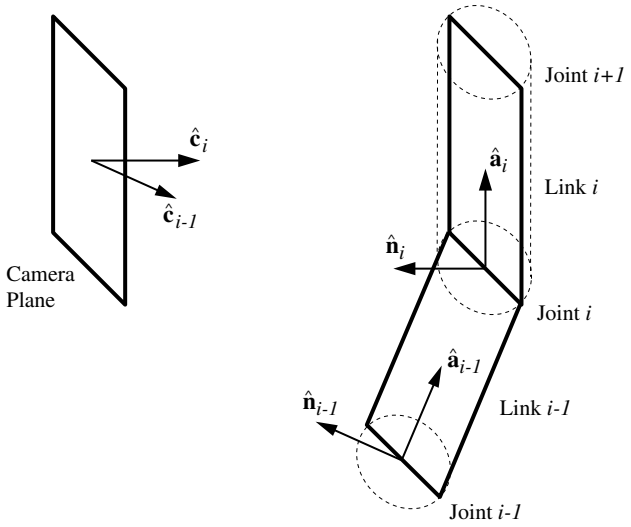

Fig. 1. Schema of the projection of the template attached to link i along the camera axis $\hat{\mathbf{a}}_i$. $\hat{\mathbf{n}}_i$ is constrained to lie in the plane spanned by $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{c}}_i$ so that the template faces the camera.

pixel measurements which models the appearance of the link from a particular viewpoint. This is illustrated in Figure 1.

Let $\mathbf{T}_i$ be the template plane for link i with unit normal vector $\hat{\mathbf{n}}_i$. Let $\hat{\mathbf{a}}_i$ be the direction of the center axis for the link and let $\hat{\mathbf{c}}_i$ denote the viewing direction. $\hat{\mathbf{c}}_i$ is directed along the line connecting the ith joint center with the center of projection of the camera. The geometry is illustrated in Figure 1. In order for $\mathbf{T}_i$ to describe the appearance of the link from any viewing direction $\hat{\mathbf{c}}_i$, we require $\hat{\mathbf{a}}_i \in \mathbf{T}_i$ (i.e., $\hat{\mathbf{a}}_i \perp \hat{\mathbf{n}}_i$). Note however that there is one viewing direction for which the model is not defined: $\hat{\mathbf{a}}_i = \hat{\mathbf{c}}_i$. In this case the camera is imaging the cross-section of the link and a view-based model with the topology of the circle is not applicable. In practice, non-terminal links will be occluded under these viewing conditions, but it is an issue for the terminal links (i.e., the fingertips). One strategy is to employ a view-based representation based on the topology of the sphere rather than the topology of the circle for these links. We will describe this type of model in more detail later in the context of an appearance model for the base link.

For a particular choice of $\hat{\mathbf{c}}_i \neq \hat{\mathbf{a}}_i$, $\hat{\mathbf{n}}_i$ is chosen so as to maximize $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{c}}_i$. Thus $\hat{\mathbf{n}}_i$ should lie in the plane spanned by $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{c}}_i$ (i.e., $\hat{\mathbf{n}}_i \perp \hat{\mathbf{a}}_i \times \hat{\mathbf{c}}_i$). An orthogonal basis for this plane is given by $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{a}}_i \times \hat{\mathbf{c}}_i \times \hat{\mathbf{a}}_i$. Expanding $\hat{\mathbf{c}}_i$ with respect to this basis, we have $\hat{\mathbf{n}}_i = \arg\max_{\mathbf{n}}[(\alpha\hat{\mathbf{a}}_i + \beta(\hat{\mathbf{a}}_i \times \hat{\mathbf{c}}_i \times \hat{\mathbf{a}}_i)) \cdot \mathbf{n}] = \hat{\mathbf{a}}_i \times \hat{\mathbf{c}}_i \times \hat{\mathbf{a}}_i$. Note that when $\hat{\mathbf{c}}_i \perp \hat{\mathbf{a}}_i \rightarrow \hat{\mathbf{n}}_i = \hat{\mathbf{c}}_i$. Intuitively this means that while the axis of the template is constrained to connect the joint centers, the template plane will orient itself around this axis in such a way to maximize the area facing the camera.

From a geometric viewpoint, the template plane approximation is reasonable when the depth variation along the link is small compared to the distance to the camera. This is almost always the case for the common scenarios of hand and figure tracking. In addition, if a link is approximately cylindrical and uniform in texture, then only a single template is required to describe its appearance from all viewing directions. This is often true for finger phalanges and limbs.

The template plane model describes the two most important components of the motion flow field induced by a link in a kinematic chain: the foreshortening effect as the link is inclined towards or away from the camera, and the in-plane component of the rotation and translation of the link with respect to the camera. What the template plane model fails to capture is motion due to a rotation around the vector $\hat{\mathbf{a}}_i$ (typically the axis of symmetry for an approximately symmetric link). This motion is cancelled by the constraints which keep the template plane oriented towards the camera.

The template plane model would be inaccurate for links whose shading patterns or occluding limb contours change in a complex way as a consequence of rotation around the link axis. In these situations, we can retain the locally planar, view-based nature of our framework by using the affine or projective motion models described in Bergen et al. (1992).

The main benefit of the template plane model for links with simple shapes is that it reduces the potentially complex relationship between surface points on the link and their image plane projections to a homography between planes. In this sense, it is a natural extension of the homographic motion model to articulated objects. From this perspective, the (possibly restricted) homographic flows for a link are parametrized by its degrees of freedom (DOFs) in the chain. Moreover, a single joint in a kinematic chain will influence the flows for all of the links it can actuate.

There are three practical benefits of this template plane representation, as follows.

- It greatly simplifies the computation of both the forward visual kinematics (which specify a generative appearance model for the object) and the visual Jacobian computation for image flow (a key step in gradient-based tracking).
- It simplifies the initialization of link appearance models from image data. Given a segmented image patch, a simple planar warp creates the texture map.
- Under a scaled orthographic projection model, the homographic flow for a link k is completely determined by the instantaneous motion of the vector $\hat{\mathbf{a}}_i$. This greatly simplifies the analysis of ambiguities and singularities by reducing the articulated object to its skeleton.

In our framework, each template has a support map which describes the active pixel locations. This support map is initialized when the templates are acquired, typically by segmenting the pixels for each link using a rough rectangular bounding box. Another possibility is to use the expectation–maximization algorithm to estimate the support maps based on the articulated motion, as described in Rowley and Rehg (1997) and Bregler and Malik (1998). A more complex and accurate appearance model based on texture-mapped cylinders is described in Sidenbladh, De la Torre, and Black (2000). A potential benefit of this model is that it can represent the flow field component that results from rotations around the cylinder axis. In practice, however, we have found that the planar template model performs quite well on many typical sequences, as we will demonstrate in Section 6.

2.2. Forward Visual Kinematics of the Template Plane Model

We now derive the forward visual kinematics for links in a kinematic chain using the planar template model. The visual kinematics have three components. The first is the standard serial kinematics that describe the 3D displacement of the links as a function of the joint angles along the kinematic chain. The second component, which is specific to visual tracking, is the camera transform that projects pixels in template plane coordinates into the image plane. The third component is the likelihood function, which specifies the measurements that can be obtained from the image pixels.

Any standard kinematics formulation can be employed to compute the 3D displacement of the links. From the basic forward kinematics we extract the locations of the joint centers for the links, with respect to the camera coordinate frame. For example, using the standard Denavit–Hartenberg (DH) notation, the transform from the joint center at link i to the joint center at link $i+1$ is $\mathbf{T}_i^{i+1}(\theta_i, d_i, a_i, \alpha_i)$ (Spong 1989). Let $\mathbf{d}_c^i$ be the location of the joint center for link i in camera coordinates. It is given by the translation component of the product of displacements $\mathbf{T}_c^i = \mathbf{T}_c^b \mathbf{T}_b^1 \ldots \mathbf{T}_{i-1}^i$. Here $\mathbf{T}_c^b$ maps from the camera frame to the base frame of the chain. Note that we number the links in a chain in increasing order, starting at the base. Thus $j < i$ implies that joint j actuates link i.

For consistency in treating terminal links, we define a virtual terminal joint center which is located at the end of the central axis of the terminal link. For example, a simple planar model of the finger might have three links and three physical joint centers.[3] To this we add an additional virtual joint center (with no DOFs) at the tip of the finger, so that each link lies between two joint centers. The preceding analysis applies to links in a kinematic chain, such as the phalanges and limbs of the body. For base links, such as the palm and torso, we employ a slightly different approach which will be described below.

Given the vectors $\mathbf{d}_c^i$ and $\mathbf{d}_c^{i+1}$ computed using standard forward kinematics, the forward visual kinematic function for link i is computed as follows

$$\hat{\mathbf{a}}_i = \frac{\mathbf{d}_c^{i+1} - \mathbf{d}_c^i}{\|\mathbf{d}_c^{i+1} - \mathbf{d}_c^i\|} \quad (1)$$

$$\hat{\mathbf{c}}_i = \frac{\mathbf{d}_c^i}{\|\mathbf{d}_c^i\|} \quad (2)$$

$$\hat{\mathbf{b}}_i = \hat{\mathbf{a}}_i \times \hat{\mathbf{c}}_i \quad (3)$$

$$\mathbf{H}^i(u, v) = \mathbf{d}_c^i + u\hat{\mathbf{a}}_i + v\hat{\mathbf{b}}_i \quad (4)$$

$$\mathbf{F}^i(u, v) = \mathbf{P}\mathbf{H}^i(u, v) \quad (5)$$

where $\mathbf{H}^i$ traces out the template plane for link i in three dimensions as a function of the template coordinates (u, v), $\mathbf{P}$ is the camera projection matrix, and $\mathbf{F}^i$ maps a point (u, v) in template plane i to a point (x, y) in the image plane. Equations (1)–(5) give a procedure for projecting the template plane into the image plane as a function of the kinematic parameters. The final component of the forward kinematics is the likelihood function, which we will discuss in the context of forming the visual Jacobian.

2.3. Visual Jacobian of the Template Plane Model

The visual Jacobian has three components. The first component maps joint velocities into the 3D velocities of points in the template plane. The second component projects these 3D velocities according to the camera projection matrix to obtain image flow fields. This is a linear transformation under the orthographic conditions which we assume for the purpose of this paper. Taken together, these first two components describe the instantaneous movement of the template plane pixels under the action of the joints. The third component is derived from the likelihood function and connects pixel motion to image

3. Moving out from the palm, they would be the metacarpophalangeal joint, proximal phalange, proximal interphalangeal joint, middle phalange, distal interphalangeal joint, and distal phalange.

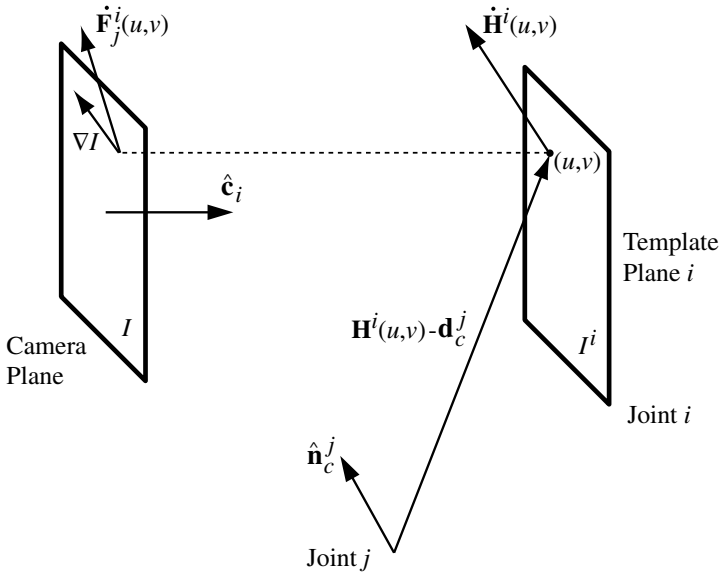

Fig. 2. Formation of the visual Jacobian. Pixel (u, v) in template i has 3D position $\mathbf{H}^i(u, v)$. An instantaneous rotation by $\dot{\theta}_j$ at joint j induces the 3D velocity $\dot{\mathbf{H}}^i(u, v)$. Projecting this velocity into the image plane yields the flow vector $\dot{\mathbf{F}}^i_j(u, v)$. For an SSD likelihood function, the change in the residual $\dot{R}_j$ is given by projection of the flow along the image gradient.

measurements. The most basic step in constructing the Jacobian is to compute the image plane velocity of a pixel with template coordinates (u, v) in template plane i due to the action of joint j, with $j < i$. The geometry is illustrated in Figure 2.

From the definition of angular velocity and eqs. 4 and 5 we have

$$\dot{\mathbf{F}}^i_j(u, v) = \mathbf{P}\frac{\partial}{\partial t}\mathbf{H}^i(u, v) \quad (6)$$

$$= \mathbf{P}(\dot{\theta}_j\hat{\mathbf{n}}^j_c \times (\mathbf{H}^i(u, v) - \mathbf{d}^j_c)) \quad (7)$$

$$\mathbf{f}^i_j(u, v) = \frac{\partial}{\partial\dot{\theta}_j}\dot{\mathbf{F}}^i_j(u, v) \quad (8)$$

$$= \mathbf{P}(\hat{\mathbf{n}}^j_c \times (\mathbf{H}^i(u, v) - \mathbf{d}^j_c)) \quad (9)$$

where $\hat{\mathbf{n}}^j_c$ is the joint axis for joint j in camera coordinates and $\dot{\theta}_j$ is the angular velocity of joint j. The function $\dot{\mathbf{F}}^i_j(u, v)$ gives the image flow at $\mathbf{F}_i(u, v)$ due to the action of joint j. The flow Jacobian $\mathbf{f}^i_j(u, v)$ is a 2-vector from which the full Jacobian can be constructed.

We now illustrate the Jacobian computation for two representative likelihood functions: point correspondences and sum of squared differences (SSD) for pixel values. The general form for the likelihood function is $p(\mathbf{z}|\mathbf{x}) = k\exp(-\frac{1}{2}\mathbf{R}^{\mathrm{T}}\Sigma^{-1}\mathbf{R})$, where $\mathbf{R}(\mathbf{z}, \mathbf{x})$ is the residual error vector between the input image and state estimate, Σ is a covariance weighting factor, and k is a normalizing constant.

In the case of point correspondences, $\mathbf{R}$ is a vector of distances in the image plane between a set of point measurements $\{\mathbf{p}_k\}$ and their corresponding points (in template coordinates) on the articulated model. Let the feature point $\mathbf{p}_k$ in the input image correspond to point (u_k, v_k) in template plane m_k. The residual for this point measurement and its associated Jacobian is given by

$$R(k) = \mathbf{F}^{m_k}(u_k, v_k) - \mathbf{p}_k \quad (10)$$

$$\dot{R}^{m_k}_j(k) = \dot{\mathbf{F}}^{m_k}_j(u_k, v_k) \quad (11)$$

$$\mathbf{J}^{m_k}_j(k) = \mathbf{f}^{m_k}_j(u_k, v_k) \quad (12)$$

where $\dot{R}^{m_k}_j(k)$ gives the change in the residual for point k under the action of joint j. The jth column in the full Jacobian matrix (corresponding to joint j) is constructed by stacking together M of the Jacobian components $\mathbf{J}^{m_k}_j(k)$, one for each of the feature points in the input image. Feature points which correspond to template points that are not actuated by link j will have an entry of zero.

In the case of SSD likelihoods, $\mathbf{R}$ is a vector of pixel differences formed by comparing corresponding pixels in the image and in the template models. The residual and Jacobian components for pixel k can be written

$$R(k) = I(\mathbf{F}^{m_k}(u_k, v_k)) - I^{m_k}(u_k, v_k) \quad (13)$$

$$\dot{R}^{m_k}_j(k) = \nabla\mathbf{I}^{\mathrm{T}}\dot{\mathbf{F}}^{m_k}_j(u_k, v_k) \quad (14)$$

$$\mathbf{J}^{m_k}_j(k) = \nabla\mathbf{I}^{\mathrm{T}}\mathbf{f}^{m_k}_j(u_k, v_k) \quad (15)$$

where $I(x, y)$ is the current image frame and $I^{m_k}(u_k, v_k)$ is the corresponding template location for pixel k. The last equation follows from eq. 9.

Column j in the full Jacobian, $\mathbf{J}$, is constructed by stacking together M of the Jacobian components from eq. 15, one for each pixel in each template. If it were constructed explicitly, the Jacobian would have M rows and N columns, where N is the number of states in the kinematic model and M is the number of features (e.g., pixels) in the planar appearance model. In practice, it is more efficient to directly construct the smaller matrices involved in estimation directly, such as $\mathbf{J}^{\mathrm{T}}\mathbf{J}$ and $\mathbf{J}^{\mathrm{T}}\mathbf{R}$, by scanning the template pixels in chain and raster order.

2.4. Template Plane Model for Base Links

It is common for a branched kinematic chain to have a base link with more complex shape than the individual links in the chain. The torso of the figure and the palm of the hand are two examples. While the motion of the base can often be recovered by tracking with measurements from the chain links only, it may be useful or necessary to include the base link in the likelihood model.

The DOFs at the base are the six degrees of rotation and translation of a rigid body. Computing the Jacobian contributions for these states due to the links in the chain is straightforward. The translation component affects all links equally through the base. The spatial rotation can always be represented as a rotation around an axis. Thus, the same Jacobian analysis for revolute joints applies in this case as well. In practice we use quaternions (McCarthy 1990) to represent the rotation of the base link.

There are many possible choices for the appearance model of the base link, which can be treated as in any standard rigid body tracking problem. The approach which might be closest in spirit to our treatment of the links would be to employ an affine or perspective flow approximation, following Bergen (1992).

3. Ambiguities in Visual Tracking with 3D Kinematic Models

A key issue in the performance of a 3D tracking system is the number of camera viewpoints which are available simultaneously. For example, commercial optical motion capture systems can provide extremely accurate reconstructions of 3D motion. These systems benefit from markers on the actor's clothing which provide good features for tracking, and multiple camera viewpoints which make it possible to localize marker positions in 3D via triangulation. One of the earliest examples of multi-view motion data is the Muybridge plates, for which markerless 3D reconstructions have been obtained (Bregler and Malik 1998).

In many interesting situations, however, only a single camera view of an articulated motion may be available. For example, the vast corpus of movie and television footage consists primarily of monocular sequences of human motion. Much of this footage is of historical or artistic significance, and it would be extremely interesting to obtain 3D reconstructions of famous dance sequences or sports events. In many of these cases, monocular video is the only archival record available. We have referred to this as the problem of *motion capture from movies* (Rehg and Morris 1997; Rehg 2000).

The 3D reconstruction of human motion from unconstrained monocular video is an extremely challenging problem, which is significantly harder than multi-camera tracking under laboratory conditions. The constraints on motion that derive from a 3D kinematic model have proven to be important in attempting monocular 3D tracking. In this section we will demonstrate that significant ambiguities still remain even when 3D kinematic models are available. We approach this problem from two inter-related viewpoints. The first viewpoint is a natural consequence of the well-known reflective ambiguity under orthographic projection. The second viewpoint comes from studying the singularities of the Jacobian of the forward visual kinematics.

3.1. Reflective Ambiguities in 3D Tracking

In analyzing ambiguities of 3D reconstruction for articulated objects, it would be convenient to find an abstraction for the shapes of the links. Obtaining useful global results in kinematic analysis for objects with varying joint topologies is already challenging. Including the impact of arbitrary link shape seems daunting. An intuitively obvious abstraction would be to replace each link with its central axis, the 3D line segment connecting its two joint centers. This effectively reduces the articulated object to its "skeleton". We now demonstrate that this intuitive simplification is mathematically justifiable in the case of template plane appearance models.

PROPOSITION 1. The image plane motion of pixels under the template plane appearance model for an articulated object is completely determined by the 3D position of the joint centers and joint axes with respect to the camera. The only exception occurs when the central axis for the link is parallel to the viewing direction, and the template model is undefined.

Proof. It is enough to demonstrate the proposition for a single link i under the action of a joint j. By substituting eqs. (1)–(4) into eq. (5) we can make the dependency on the joint centers $\mathbf{d}_c^i$ and $\mathbf{d}_c^{i+1}$ explicit

$$\mathbf{F}_i(\mathbf{d}_c^i, \mathbf{d}_c^{i+1}, u, v) = \mathbf{P}[\mathbf{d}_c^i + \frac{u}{\rho}(\mathbf{d}_c^{i+1} - \mathbf{d}_c^i) + \frac{v}{\rho\|\mathbf{d}_c^i\|}(\mathbf{d}_c^{i+1} \times \mathbf{d}_c^i)] \tag{16}$$

where we define $\rho = \|\mathbf{d}_c^{i+1} - \mathbf{d}_c^i\|$. Thus the image plane position of an arbitrary template plane point (u, v) is determined by $\mathbf{d}_c^i$ and $\mathbf{d}_c^{i+1}$. Similarly, we can modify eq. (9) to show the dependence of $\mathbf{H}_i$ on $\mathbf{d}_c^i$ and $\mathbf{d}_c^{i+1}$:

$$\mathbf{f}_j^i(\mathbf{d}_c^i, \mathbf{d}_c^{i+1}, \mathbf{d}_c^j, \hat{\mathbf{n}}_c^j, u, v) = \mathbf{P}(\hat{\mathbf{n}}_c^j \times (\mathbf{H}_i(\mathbf{d}_c^i, \mathbf{d}_c^{i+1}, u, v) - \mathbf{d}_c^j). \tag{17}$$

The direction of flow at an arbitrary template plane point (u, v) is determined by $\mathbf{d}_c^i$, $\mathbf{d}_c^{i+1}$, $\mathbf{d}_c^j$, and $\hat{\mathbf{n}}_c^j$, while the magnitude is proportional to $\dot{\theta}_j$. The exception for cases where the central axis is parallel to the viewing direction (i.e., $\hat{\mathbf{a}}_i = \hat{\mathbf{c}}_i$) is necessary since $\hat{\mathbf{n}}_i$ is not uniquely defined in that case (see Section 2.1). This establishes the result.

Proposition 1 shows that given the 3D joint centers we can almost always obtain the template appearance. However, given only the template appearance there is a common ambiguity in determining 3D positions. The reflective ambiguity under orthographic projection leads to two possible solutions for the 3D reconstruction of a "skeletal" link from its camera projection. The basic ambiguity is illustrated in Figure 3(a). This ambiguity has significant consequences for tracking with 3D models. While the two solutions, f and g, are indistinguishable from the standpoint of a single point measurement and its velocity, they result in different kinematic constraints on the motion. This is illustrated by the two velocity vectors in the figure. Therefore it is important to identify the correct solution (i.e., to distinguish between f and g) in order to exploit the 3D constraint for tracking.

It is possible that other sources of information, such as predictions from a dynamic model, could help in distinguishing between these two cases. However, it is difficult to obtain reliable predictions given the complexity of human movement and the 30 Hz sampling rates of conventional monocular video. Alternatively, a perspective projection model could in principle resolve the ambiguity. However, perspective effects at this scale are usually difficult to resolve in the presence of noise, given the camera geometries which are common in figure and hand tracking. This is particularly true given the fact that measurements of the projections of the joint centers can only be obtained indirectly through contour and texture cues.

One solution, proposed in Shimada et al. (1998), is to use multiple hypothesis tracking techniques to maintain separate estimates of the different possible 3D configurations during tracking. The idea is that over time certain solutions will be ruled out by the global motion and constraints such as joint angle limits. However, for a kinematic chain of N links there will be 2^N possible solutions for the 3D configuration of the chain. This leads to significant complexity for an on-line tracking algorithm.

Another aspect of the reflective ambiguity is illustrated in Figure 3(b), which shows a configuration of the link at which the two ambiguous solutions merge into one. As we will show in Section 3.2, these bifurcation points correspond to *singularities* in the visual Jacobian which maps state velocities into image velocities. In this singular configuration, the motion of the link is directed along the camera axis and as a consequence its projection into the image plane is zero. Singularities can cause difficulties for standard gradient-based 3D tracking algorithms which are based on inverting the visual Jacobian.

An alternative approach, which we develop in detail in Section 4, is to modify the kinematic representation so it encodes only the information contained in the image measurements. This approach defers the question of 3D reconstruction to a later stage of analysis. Figure 3(c) illustrates this representation for the single link case. By representing only the image plane projection of the link we can avoid the ambiguity in 3D reconstruction. This model for a link in a kinematic chain is analogous to the camera model proposed by Koenderink and van Doorn (1991) for affine reconstruction.

The affine link representation can be beneficial when combined with stochastic tracking algorithms such as particle filters (Isard and Blake 1996; MacCormick and Blake 1999; Deutscher, Blake, and Reid 2000). Particle filters address background clutter and other sources of ambiguity during tracking through a nonparametric representation of a multimodal density function. Figure 3(d) illustrates the application of a particle filter to the single link example. In the case of a 3D state space, the reflective ambiguity is lumped together with all of the other sources of ambiguity. This results in the two sets of particles labeled f' and g'. In contrast, the affine representation requires only the single set h', which addresses the noise and clutter in the image data without attempting to extrapolate it into 3D.

The advantage of the affine model in Figure 3(c) is its lack of ambiguities, and its disadvantage is the loss of 3D constraints on the image motion. However, by tracking with an affine model we can aggregate measurements across an image sequence before attempting 3D reconstruction. This could make it easier to reject ambiguous hypotheses, leading to more efficient 3D reconstruction. It also mirrors the decomposition of 2D tracking followed by 3D reconstruction which is common in structure-from-motion algorithms (Tomasi and Kanade 1992).

3.2. Visual Singularities in 3D Tracking

The simplest non-probabilistic tracking algorithm consists of using a nonlinear least-squares solver, such as the Levenberg–Marquardt algorithm (Dennis and Schnabel 1983), to minimize the error measure $E(\mathbf{q}) = \frac{1}{2}\mathbf{R}^{\mathrm{T}}\mathbf{R}$, where the residual vector $\mathbf{R}$ is formed, for example, using eq. (10) or eq. (13). In this case the state vector $\mathbf{q}$ holds the joint angles and base link displacement for a branched kinematic chain. The update step in the Levenberg–Marquardt algorithm is given by

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \mathbf{dq}_k = \mathbf{q}_{k-1} - (\mathbf{J}^{\mathrm{T}}\mathbf{J} + \Lambda)^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{R}, \tag{18}$$

where Λ is a diagonal stabilizing matrix. The structure of the Jacobian $\mathbf{J}$ is governed by eq. (9). It describes the projection of a 3D velocity vector into the image plane. In the case of an SSD residual, this projected vector is in turn projected onto

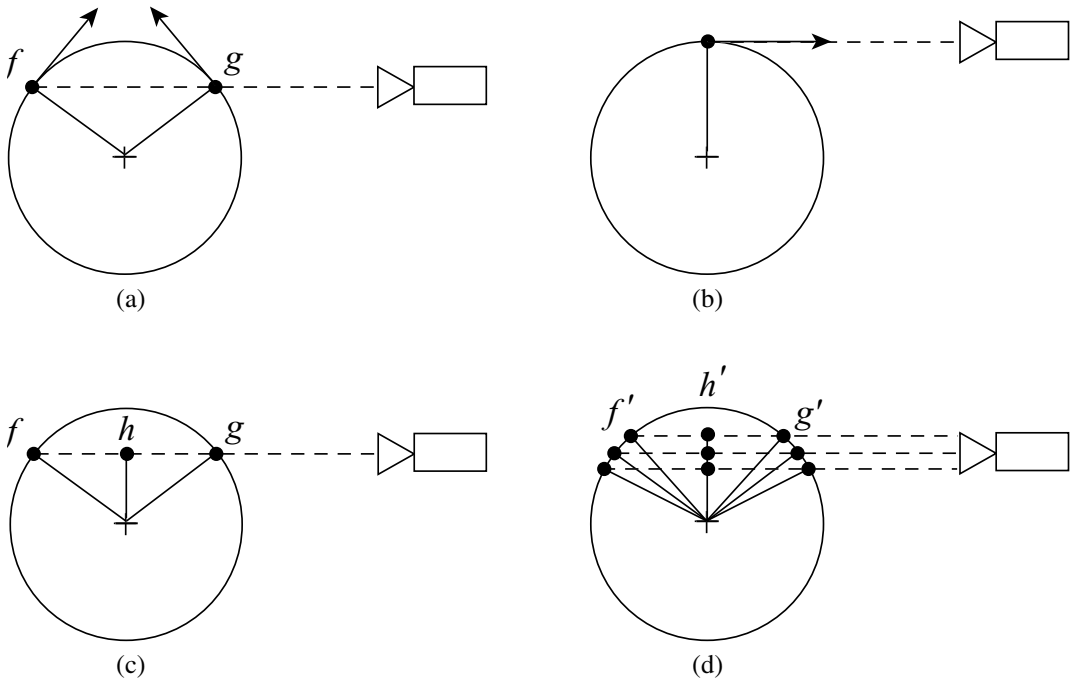

Fig. 3. Ambiguity in estimating 3D pose of a single link given its orthographic projection. (a) The points f and g produce the same image measurements but impose different kinematic constraints (arrows). (b) A singularity occurs when the link velocity projects to zero. (c) An affine kinematic model gives the point h which encodes the image measurement. (d) Particle representations produce the point sets f' and g' in three dimensions, but requires only the single set h' under the affine model.

the local image gradient, as given by eq. (15) and illustrated in Figure 2.

From the figure it is clear that residual $R(i)$ can provide no information about q_j if $\mathbf{J}_j^{k_i}(i)$ is zero. This would be the case if the 3D point velocity $\dot{\mathbf{F}}_j^{k_i}(u_i, v_i)$ were directed along the optical axis (as in Figure 3(b) for example), or (in the SSD residual case) if its non-zero projection acts perpendicular to the image gradient.

The complete Jacobian $\mathbf{J}$ is formed by stacking up the $\mathbf{J}_j^{k_i}(i)$ terms from each measurement. The result is a linear map from state space to residual space. The nullspaces of this mapping provide fundamental insight into its properties. The left nullspace of the Jacobian, $\mathcal{N}(\mathbf{J}^{\mathrm{T}})$, defines the constraints inherent in the kinematic model. Residual velocities in the left nullspace, $\dot{R} \subseteq \mathcal{N}(\mathbf{J}^{\mathrm{T}})$, are excluded. An empty left nullspace indicates that the kinematics do not constrain the motion. In tracking there will typically be more image measurements than parameters, $M > N$, resulting in a non-empty left nullspace and $\mathrm{rank}(\mathbf{J}) = N$.

The right nullspace of the residual Jacobian defines the *visual singularities* of the articulated object. State velocities in the right nullspace, $\dot{\mathbf{q}} \subseteq \mathcal{N}(\mathbf{J})$, do not affect the residual $\dot{\mathbf{R}}$ and so are termed *singular directions*. The right nullspace is non-zero only when the Jacobian has lost rank, i.e., $\mathrm{rank}(\mathbf{J}) < N$. It follows that if the projected velocities due to the action of any joint j are zero for all template pixels, then there will be a zero column in the Jacobian, leading to a loss of rank.

3.2.1. An Example of a Visual Singularity

We now present two simple examples which illustrate the properties of the visual Jacobian and the existence of visual singularities. Figure 4(a) shows a one-link revolute planar manipulator with a single DOF θ. The residual Jacobian for the end-point feature is defined by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ -\sin(\theta) \end{bmatrix} \begin{bmatrix} \dot{\theta} \end{bmatrix}, \tag{19}$$

assuming that the camera and joint axes are parallel. The kinematic constraint is given by the left nullspace: $\dot{R}_c = \begin{bmatrix} \sin(\theta) & \cos(\theta) \end{bmatrix}^{\mathrm{T}}$. The right nullspace is empty and there are no observability singularities.

Next consider the manipulator in Figure 4(b), formed by adding an additional DOF, ϕ, to Figure 4(a), which allows the link plane to tilt out of the page. With the same point feature and camera viewpoint we have

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\sin(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \\ 0 & -\sin(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix}. \tag{20}$$

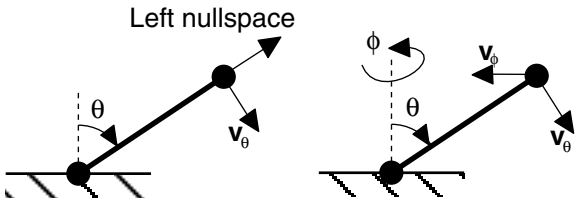

Fig. 4. Examples of (a) 1-DOF and (b) 2-DOF manipulators.

Singularities now occur when $\sin(\phi) = 0$ and also when $\sin(\theta) = 0$. In both cases the singular direction is $\dot{\mathbf{q}} = [1\ 0]^{\mathrm{T}}$, implying that changes in ϕ cannot be recovered in these two configurations.

Singularities impact visual tracking through their effect on error minimization. Consider tracking the model of Figure 4(b) using the Levenberg–Marquardt update step:

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \mathbf{dq}_k = \mathbf{q}_{k-1} - (\mathbf{J}^{\mathrm{T}}\mathbf{J} + \Lambda)^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{R}. \tag{21}$$

At the singularity $\sin(\phi) = 0$, the update step for all trajectories has the form $\mathbf{dq}_k = [0\ C]$, implying that no updates to ϕ will occur regardless of the measured feature motion. This singularity arises physically when the link rotates through the plane parallel to the image plane, resulting in a point velocity in the direction of the camera axis.

Figure 5(a) illustrates the practical implications of singularities for tracker performance. The stair-stepped curve corresponds to discrete steps in ϕ in a simulation of the 2-DOF example model. In this example, the arm is planar with a randomly textured template model. The solid curve shows the state estimates produced by eq. (21) as a function of the number of iterations of the solver. The loss of useful gradient information and resulting slow convergence of the estimator as the trajectory approaches the point $\phi = 0$ is symptomatic of tracking near singularities. In this example, the singular state was never reached and the tracker continued in a direction opposite the true motion, as a consequence of the usual reflective ambiguity under orthographic projection (shown by the dashed line). Perspective projection also suffers from this ambiguity for small motions.

These examples illustrate the significant implications of singularities for visual tracking. If the search for feature measurements is driven by prediction from the state estimates, singularities could result in losing track of the target altogether. Even when feature correspondences are always available, such as when markers are attached to the object, the solver will slow down dramatically near singularities, since each step has only a small effect on the residual. This is analogous to the effect of classical kinematic singularities in robotic manipulators (Nakamura 1991)—manipulator control near singularities may require arbitrarily large forces; here tracking near singularities may require arbitrarily large numbers of iterations!

It would be useful to obtain general conditions under which singularities can arise in tracking with 3D kinematic models. This is a challenging task due to the high dimensionality and nonlinearity of kinematic models. Attempts have been made to classify the singularities in robot manipulators from the standpoint of both manipulator design (Pai 1988) and visual-servoing control (Sharma and Hutchinson 1997).

From eq. (9) we can derive some sufficient conditions for the existence of visual singularities for a template plane i and joint j:

$$\hat{\mathbf{n}}_i \parallel \hat{\mathbf{c}}_i, \quad (\mathbf{d}_c^i - \mathbf{d}_c^j) \perp \hat{\mathbf{c}}_i, \quad \hat{\mathbf{n}}_c^i \perp \hat{\mathbf{c}}_i. \tag{22}$$

Whether these are necessary conditions is a topic for future research.

We can make two observations about eq. (22). First, use of the planar appearance model makes it possible to greatly simplify the singularity analysis for a link. More complex link shape models may be less likely to generate visual singularities, but whether this is true in practice is a subject for further investigation. Measuring subtle differences in velocity due to subtle differences in shape can be difficult, particularly in unconstrained video footage such as movies. Secondly, since this analysis stems fundamentally from the orthographic projection model and the definition of angular velocity, it does not depend upon any particular parametrization of the kinematic DOFs. It should therefore apply to any specific kinematic representation used in tracking.

4. A 2D Kinematic Model for Articulated Object Registration

The previous section outlined the conditions under which singularities can occur in 3D kinematic models. Singularities have two implications for 3D tracking with a single video source. First, some additional source of information is required in order to estimate the unobservable parts of the state space during singular configurations. For example, assumptions about object dynamics could be used in a Kalman filter framework to extrapolate an estimated state trajectory across a singular point. Secondly, the utility of the kinematic model for image registration is reduced, since it will not always supply a useful constraint on pixel motion.

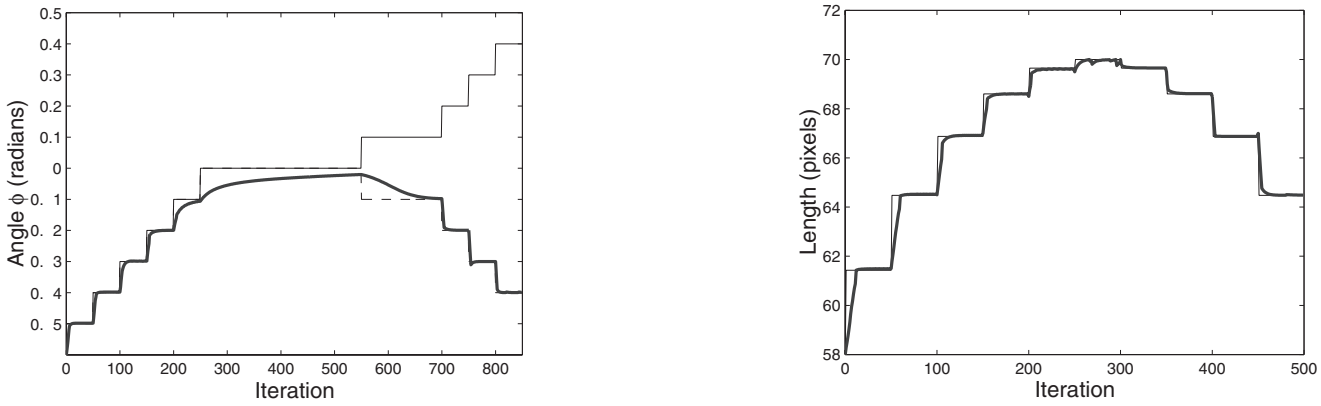

Fig. 5. Singularity example. (a) Tracking the 3D manipulator in example 2 through a singular point along the singular direction. While the true angle ϕ continues to increase, the tracker loses track near the singularity and then picks up an ambiguous path. (b) The 2D tracker from Section 4 is applied to the same motion as in (a), but here extension length rather than angle is recovered, and this correctly increases and decreases without change in damping.

It is important here to distinguish two separate goals: a registration objective in which the model projections are aligned with image pixels, and a reconstruction goal in which the state trajectory for a 3D kinematic model is estimated. For some applications, such as gesture recognition or video editing (Rehg, Kang, and Cham 2000), registration may be all that is required. In other applications such as motion capture it is desirable to reconstruct the 3D motion along with the kinematic model parameters. Once the images have been registered, 3D reconstruction can be cast as a batch estimation problem, since the registration step gives the complete correspondence between model points and image coordinates in each frame. The batch nature of the formulation is well suited to our application of motion capture from movies, and makes it possible to enforce smoothness constraints in both time directions, improving the quality of the estimates. In Section 4.4 we describe this approach to reconstruction in more detail.

The remainder of this section focuses on the registration step. Although we do not want to employ the full 3D model, we would like to employ the strongest possible kinematic constraints so as to improve robustness to image noise. We will see that a novel 2D SPM formed by "projecting" the 3D model into the image plane provides a useful constraint for registration.

4.1. The Scaled Prismatic Model

The 2D SPM acts in a plane parallel to the plane of the camera and simulates the image motion of the 3D model. Links have the same connectivity as in the 3D model, rotate around their base joint on an axis perpendicular to the plane, and scale uniformly along their length. Each link is thus represented as a line segment having two parameters: its angle of rotation θ_i and length d_i along its direction $\hat{\mathbf{a}}_i$. As in the 3D case a template is attached to each link which rotates and scales with the link. Figure 6 shows both of these parameters for a link in the 2D SPM.

In this section we briefly derive the kinematics of this model class, we show that it can capture the projected motion of a 3D figure, and then we show that it is precisely in the cases where the 3D model suffers from singularities that the 2D SPM behaves well.

4.2. Kinematic Equations for a Chain of Links

The residual velocity can be expressed as the sum of the Jacobian columns times their corresponding state parameter velocity: $\dot{\mathbf{R}} = \sum_i \mathbf{J}_i \dot{q}_i$. Hence we can calculate individual Jacobian columns for each state variable, q_i independently and then combine them. Since a column of the Jacobian, $\mathbf{J}_i$, maps the state velocity $\dot{q}_i$ to a residual velocity, by finding the residual velocity in terms of this state we can obtain an expression for $\mathbf{J}_i$.

The components of the Jacobian can be obtained as follows. The first two parameters, x_0 and y_0, specify the origin of the base link. As this point moves, all positions on the figure move in the same manner. Hence a pixel p will have velocity $\mathbf{v}_p = \dot{\mathbf{r}}_0$, implying that the Jacobian for this pixel is a 2×2 identity, $\mathbf{I}_2$. The two columns of the Jacobian relating each pixel residual to the velocity of these parameters are formed by stacking $\mathbf{I}_2$ in a column for each pixel.

Next we consider the Jacobian component due to a rotation of a joint. If $q_i = \theta$ is the angle of a revolute joint shown in Figure 6(a), it will contribute an angular velocity component

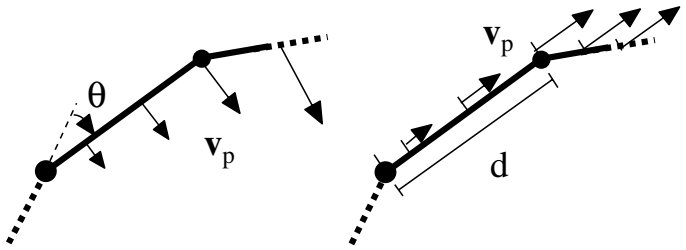

Fig. 6. 2D SPM chain showing residual velocities due to state velocities: (a) $\dot{q}_i = \dot{\theta}$, and (b) $\dot{q}_i = \dot{d}$.

to links further along the chain given by $\omega = \dot{q}\mathbf{w}$. Here $\mathbf{w}$ is the axis of rotation which for the 2D model is just the z-axis. Let $\mathbf{r} = (x, y, 0)$ be a point on the manipulator chain measured with respect to the axis. The image velocity, $\mathbf{v}_p$, at point $\mathbf{r}$ resulting from this rotation is given by

$$\mathbf{v}_p = \mathbf{P}(\omega \times \mathbf{r}) = \mathbf{P}(\mathbf{w} \times \mathbf{r})\dot{q} = \mathbf{r}_{2d}\dot{q} \tag{23}$$

where the orthographic projection $\mathbf{P}$ selects the x and y components, and $\mathbf{r}_{2d} = (-y, x)$ captures the pixel position. This equation expresses the desired mapping from state velocities of axis i to image velocities of point j on link k giving the components of the column Jacobian, $\mathbf{J}_i$:

$$J_{ji} = \begin{cases} 0 & \text{links } k, \text{ where } k < i \\ \mathbf{r}_{2d} & \text{links } k, \text{ where } k \geq i \end{cases}. \tag{24}$$

The Jacobian due to extending links must also be considered. If $q_i = d$ refers to the extension of the scaled prismatic link shown in Figure 6(b), along direction $\hat{\mathbf{a}}_i$ in the plane, its derivative will contribute a velocity component to points on link i proportional to their position on the link, bq_i, where b is the fractional position of the point over the total extension q_i. The velocity component for a point, p, on the link is thus $\mathbf{v}_p = bq_i\dot{q}_i\hat{\mathbf{a}}_i$. Subsequent links, $k > i$, will be affected only by the end-point extension of the link, and so have a velocity component from this joint given by $\mathbf{v}_p = q_i\dot{q}_i\hat{\mathbf{a}}_i$. Hence the Jacobian element at point j on link k for an extension parameter, q_i, is given by

$$J_{ji} = \begin{cases} 0 & \text{links } k, \text{ where } k < i \\ bq_i\hat{\mathbf{a}}_i & \text{link } i \\ q_i\hat{\mathbf{a}}_i & \text{links } k, \text{ where } k > i \end{cases}. \tag{25}$$

Finally we find the effect of using a weak perspective model rather than the orthographic model which we have assumed up to now. We first specify a new projection matrix incorporating a global scale factor, μ, as

$$\mathbf{P} = \mu \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{26}$$

This means first that the Jacobian elements, J_{ji}, in eqs. (24) and (25) must be scaled by μ. Then if the 2D position of pixel j is $\mu\mathbf{r}_j$ with respect to the center of projection, its velocity due to rescaling is $\mathbf{r}_j\dot{\mu}$. Hence if k is the index of μ in the state vector $\mathbf{q}$, the Jacobian elements for pixel j are

$$J_{jk} = \mathbf{r}_j. \tag{27}$$

We show that, given certain modeling assumptions, the 2D SPM with the above specifications is flexible enough to represent the projected image of any 3D model in any legal configuration. We assume that a model consists of a branched chain of links connected at their end-points by revolute joints. We use the template plane model from Section 2.1 to describe link appearance. We identify the *link segment* for a link k as the 3D line segment connecting the link's joint centers and oriented in the direction of $\hat{\mathbf{a}}_k$. The 3D model specifies the figure position, the link lengths and the orientation of each revolute joint axis, while in the SPM the link lengths vary and the axis of each revolute joint is perpendicular to the image plane. The state of a 3D model is thus a 3D position of the origin of the base link plus a vector of joint angles, $\mathbf{q}_m = [\ X \quad Y \quad Z \quad \phi_1 \quad \phi_2 \quad \ldots\ ]^{\mathrm{T}}$, and the state of a 2D SPM is a 2D position plus a vector of angles and joint lengths, $\mathbf{q}_n = [\ x_0 \quad y_0 \quad \theta_1 \quad d_1 \quad \theta_2 \quad d_2 \quad \ldots\ ]^{\mathrm{T}}$. Since we are only working with end-points, scale in 2D is captured by link lengths without the need for an additional scale factor μ. Then more formally:

PROPOSITION 2. The linear projection of the link segments of a 3D kinematic model onto a plane and the assignment of revolute joints with axes perpendicular to the plane between each pair of connected links defines a many to one mapping $\mathcal{F}_M : \mathbf{M}^3 \rightarrow \mathbf{M}^2$ from the space of all 3D models $\mathbf{M}^3$ to 2D models $\mathbf{M}^2$. Furthermore, for each pair of models, $m \in \mathbf{M}^3$ and $n = \mathcal{F}_M(m)$, it defines another mapping $\mathcal{F}_S : \mathbf{Q}_m^3 \rightarrow \mathbf{Q}_n^2$ that maps every state of the 3D model $\mathbf{q}_m \in \mathbf{Q}_m^3$ to a state of the 2D SPM $\mathbf{q}_n \in \mathbf{Q}_n^2$.

Proof. Consider the graph, G, of a 3D model with each joint represented by a vertex and each link by an edge. There may be many 3D models with the same graph G since 3D joints may have multiple revolute axes. When a 3D model in any state is projected onto a plane under a linear projection, the new graph G' will have the same topology of vertices and edges, and the projected edges will remain linear. Now interpret the graph, G', drawn in the plane as each straight edge representing an

extensible link, and the intersection point of each connected pair of edges as a revolute joint. This defines a unique 2D model, and thus the mapping $\mathcal{F}_M$. The state of a 2D SPM is specified by the 2D projection, (x_0, y_0), of the 3D point (X, Y, Z), the distances in the plane between connected joints (i.e., the link lengths d_i), and the angles between links that share joints (i.e., θ_i) as illustrated in Figure 6. Now the state of the 3D model determines, through the projection, the relative positions of the vertices in two dimensions and thus the 2D state. For any distribution of vertices in the plane here must exist a 2D state $\mathbf{q}_n$ that captures it since line segments can join any two connected vertices, and any relative orientation between two line segments can be described by a single angle. There thus must exist a mapping $\mathcal{F}_S$ for all 3D states. We conclude that the 2D SPM class can capture any projected 3D kinematic model in any configuration.

4.3. Singularity Analysis of the 2D SPM

An important advantage of the SPM is the location of its singularities. In the 3D model the singularities occur in the frequently traversed region of configuration space in which links pass through the image plane. The 2D SPM has all of its rotation axes parallel to the camera axis and so never satisfies the 3D singularity conditions from eq. (22). Here we show that the SPM only has singularities when $d_i = 0$, corresponding to a 3D link pointing towards the camera, and that the singular direction is perpendicular to the entering velocity and so usually does not affect tracking.

PROPOSITION 3. Given x and y measurements of end-points of each joint in a linear chain scaled-prismatic manipulator, observability singularities occur if and only if at least one of the joint lengths is zero.

Proof. We define a state vector made of pairs of components for each link: $\mathbf{q} = \begin{bmatrix} x_0 & y_0 & \theta_1 & d_1 & \dots & \theta_n & d_n \end{bmatrix}^{\mathrm{T}}$, and the residual vector to be the error in x and y end-point positions of each link. We assume the proposition holds for an $n-1$ link manipulator with Jacobian $\mathbf{J}^{(n-1)}$ whose elements are defined as in eqs. (24) and (25). The Jacobian for the n length manipulator is given by

$$\mathbf{J}^{(n)} = \begin{bmatrix} \mathbf{J}^{(n-1)} & A \\ B & C \end{bmatrix} \tag{28}$$

where $\mathbf{J}^{(n-1)}$ is a square matrix of size $2n$. Matrix A is of size $2n \times 2$ and expresses the dependence of the nth link's parameters on the position of the other links positions and so is zero. Matrix C and its square are given as

$$C = \begin{bmatrix} \cos(\theta_T) & -d_n \sin(\theta_T) \\ \sin(\theta_T) & d_n \cos(\theta_T) \end{bmatrix}, \tag{29}$$

$$C^{\mathrm{T}}C = \begin{bmatrix} 1 & 0 \\ 0 & d_n^2 \end{bmatrix} \tag{30}$$

where $\theta_T = \sum_{i=1}^{n} \theta_i$. From this we see that C has rank two if and only if $d_n \neq 0$. If C has rank two, then the bottom two rows of $\mathbf{J}^{(n)}$ are linearly independent of all other rows and if $\mathbf{J}^{(n-1)}$ is full rank then $\mathbf{J}^{(n)}$ must have rank $2n+2$. If C or if $\mathbf{J}^{(n-1)}$ do not have full rank, then $\mathbf{J}^{(n)}$ will not have rank $2n+2$, and there will be an observability singularity. To complete the proof we need only demonstrate that the proposition applies to the base case, $n = 0$. Here the whole Jacobian is given by $\mathbf{I}_2$ and thus the proposition is proven.

A mitigating property of the 2D singularities is that, unlike in the 3D observability singularities where the singular direction is along the motion trajectory, the singular direction in the 2D case is always perpendicular to the direction in which the singularity was entered. We can see this for the single arm manipulator described by a Jacobian equal to C in eq. (29). When $d = 0$ the velocity direction is $\dot{R} = \begin{bmatrix} \cos(\theta) & sin(\theta) \end{bmatrix}^{\mathrm{T}}$, but the left nullspace is orthogonal to this by definition. Hence a manipulator will typically pass through a 2D singularity without the increased damping caused by moving along a singular direction. Only if the link enters in one direction and leaves orthogonally does the singularity obstruct tracking.

In our derivations we have only considered the case without an explicit scale factor, μ. If we include this in our vector of parameters it is easy to see that this introduces an additional observability singularity. This is because scale is redundant in specifying the 2D positions of the end-points, all of which can be specified without μ. If it is included it introduces an extra DOF, but without changing the expressivity of the model, and hence it does not affect the tracking. (See recent work on gage theory (Kanatani and Morris 2001) which has dealt with the implications of these extra DOFs.) However, when templates are included in our model, then this scale factor is no longer redundant. This is because, while it does not directly set the lengths of the templates which are free to vary, it determines their widths. Thus templates add the extra constraints to remove this DOF and the singularity.

The assumption that we have information on end-points is equivalent to assuming there is sufficient texture or edge information on the link to obtain length and direction estimates. When this assumption fails there may be more singularities for both the 3D and 2D models. While both 2D and 3D model classes can represent articulated motion, the 2D SPM provides weaker constraints. It has the two advantages of avoiding the singularities of the 3D model and relaxing the need for accurate knowledge of link lengths and joint axes which is required by the 3D model. Moreover, the 2D and 3D models are complementary in that their singularities occur in different parts of the state space.

4.4. 3D Tracking Based on 2D Registration and 3D Reconstruction

The SPM representation of articulated motion supports a decomposition of the 3D articulated tracking problem for

monocular video into the interlinked steps of 2D registration and 3D reconstruction. The primary motivation for this decomposition is illustrated in Figure 3(d). The direct application of 3D kinematic models to monocular tracking requires the joint representation of both the ambiguities resulting from 3D reconstruction and the ambiguities due to noise and background clutter that make registration challenging. In particular, each possible image plane location of the link (resulting from a possibly spurious match between the model and image) generates two possible solutions for 3D reconstruction. Decoupling these two sources of ambiguity can potentially yield more efficient probabilistic tracking algorithms, which is particularly important given the large number of states that are required by articulated objects.

One potential disadvantage of this decoupled approach is that the registration step may examine link configurations which could be ruled out by the application of 3D constraints such as the kinematic model or joint angle limits. Self-occlusion, which is the occlusion of one link of the mechanism by another during tracking, is another potential problem. In Rehg and Kanade (1995), a layered template representation was employed to track self-occluding articulated objects. Templates were ordered in depth based on predictions from a 3D kinematic model. It is straightforward to incorporate a layered template representation into the SPM framework. However, in the absence of a 3D model the template ordering will be unknown. One possibility is to approach the unknown template order through data-association techniques (Cox and Hingorani 1996). In this approach multiple trackers would be used to explore different template orderings in situations where templates overlap in the image.

Figure 7 illustrates a 3D monocular tracking framework which combines 2D registration and 3D reconstruction. The output of the registration module is a probability distribution over possible 2D alignments of an SPM model and the image sequence. This representation is then "lifted" into a distribution over 3D state trajectories by the reconstruction module. This module can employ a wide range of constraints such as a full 3D kinematic model, joint angle limits, and figure dynamics. Recently we have demonstrated results for both of these modules: probabilistic tracking using the SPM model in Cham and Rehg (1999) and non-probabilistic reconstruction of 3D state trajectories from SPM registration output in DiFranco, Cham, and Rehg (2001).

The registration and reconstruction stages in Figure 7 can be interleaved in several different ways. In the most strongly decoupled case, SPM registration of the entire video sequence could provide the inputs to a batch 3D reconstruction algorithm. More interesting perhaps would be to interleave registration and reconstruction on a per-frame basis. In this approach, a probabilistic tracking algorithm could sample from the SPM distribution in order to explore image matches. Only the most promising samples would be evaluated using the 3D kinematic model. The predicted 3D density could then be projected down into the SPM representation to initialize the next round of sampling. We plan to explore strategies of this kind in our future work.

5. Previous Work

There have been numerous papers on 3D and 2D tracking of articulated objects. However, none of these has addressed the question of singularities and their implications for tracking with a monocular video source. The 3D kinematic analysis in this paper is based primarily on our earlier work (Rehg and Kanade 1994b, 1995). The SPM model first appeared in a preliminary form in Morris and Rehg (1998).

The first works on articulated 3D tracking were by O'Rourke and Badler (1980) and Hogg (1983). They employed the classical AI techniques of constraint propagation and tree search, respectively, to estimate the state of the figure. Hogg was the first to show results for a real image sequence. A modern version of the discrete search strategy is employed by Gavrila and Davis (1996), who use a hierarchical decomposition of the state space to search for the pose of a full body 3D model.

Yamamoto and Koshikawa (1991) were the first to apply modern kinematic models and gradient-based optimization techniques to figure tracking. Their experimental results were limited to 2D motion and they did not address kinematic modeling issues in depth. The gradient-based tracking framework was extended by Rehg and Kanade (1995) to handle self-occlusions and applied to hand tracking (Rehg 1995). Early work on figure tracking by Rohr is described in Rohr 1994).

The objective function used by Yamamoto et al. compares measured image flow to the image flow predicted by the kinematic Jacobian of the figure. The same approach was explored by Pentland and Horowitz (1991) for non-rigid motion analysis, which includes an example of figure tracking. Other examples include Bregler and Malik (1998) and Sidenbladh, Black, and Fleet (2000). More recent work by Fablet and Black (2002) uses learned probability distributions over flow fields to detect and track walking motion.

A number of 3D tracking systems have used explicit shape models for the limbs, usually some form of superquadric or generalized cylinder. The system of Kakadiaris and Metaxas is a more complete example, and addresses model acquisition (Kakadiaris and Metaxas 1995) and self-occlusion handling (Kakadiaris and Metaxas 1996). Related work by Goncalves and Perona is described in Goncalves et al. (1995). Cylindrical appearance models and their acquisition are discussed in Sidenbladh, Black, and Fleet (2000); also see Delamarre and Faugeras (2001).

Gradient-based search strategies have the crucial performance advantage of exploiting information about the local shape of the objective function to select good search directions. This leads to extremely fast search performance in high-

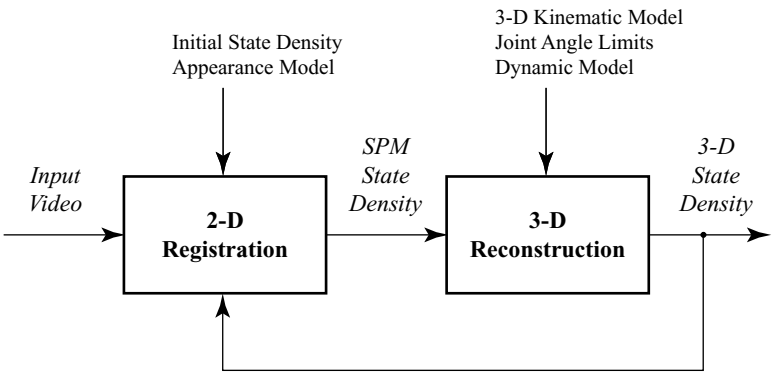

Fig. 7. Architecture for 3D monocular tracking based on 2D registration and 3D reconstruction.

dimensional state spaces. For example, Rehg and Kanade (1994b) demonstrated tracking of a 26 DOF hand model using live video. Probabilistic tracking algorithms, such as particle filters, can also exploit gradient information to improve search performance. This was first demonstrated in Heap and Hogg (1998) and Cham and Rehg (1999) and has been explored more recently in Sminchisescu and Triggs (2001).

The work of Ju, Black, and Yacoob (1996) is perhaps the closest to our 2D SPM. They model the image motion of rigid links as affine flow patches with imposed attachment constraints. Their model can be viewed as an extension of classical patch tracking techniques, which incorporates constraints between patches. In contrast, our model is derived from an explicit consideration of the effects of 3D kinematics on image motion. As a consequence, it has a more direct connection to the underlying 3D motion. We believe this property will become important in reconstructing the 3D motion of the figure from SPM measurements. The SPM also has fewer parameters than a patch-based description of flow. Other work on monocular figure tracking is described in Wachter and Nagel (1999).

Another relevant body of work from the standpoint of singularity analysis is the field of visual servo control of robot manipulators, known as visual servoing (Weiss, Sanderson, and Neuman 1987; Espiau, Chaumette, and Rives 1992; Papanikolopoulos, Khosla, and Kanade 1993). There is an analogous visual Jacobian used in visual servoing which maps changes in joint angles to changes in the position of visual features. In eye-in-hand visual servoing approaches, the joints are actuated so as to align the projection of 3D target features in the robot's environment with a 2D image model. In an alternative approach, one or more cameras in the environment acquire images of an object which is gripped by the end-effector. This latter approach is probably the closest to articulated object tracking. In both cases, visual tracking of the object or environment is used to drive servoing commands.

Singularity analysis for visual servoing problems was described in in Nelson and Khosla (1994) and Sharma and Hutchinson (1997). The context for this work was the problem of sensor placement—determining how to position cameras in a robot workspace so as to maximize the accuracy of the controller in visual servoing tasks. Both the workspace geometry and task requirements limit the range of possible end-effector motions, making optimization of the sensor location feasible.

In contrast to the controlled environment of a robot workcell, figure tracking is inherently unconstrained. In applications such as motion capture from movies, both an articulated figure and the camera itself can simultaneously undergo fairly arbitrary motion. Another practical difference is that features on the articulated object itself must be used for visual tracking, as opposed to features on a manipulated part or in the robot's environment. As a consequence, it is often challenging to obtain an adequate number of features and there are often serious problems with noise and occlusions. A final difference is the fact that, while accurate robot dynamic models are often available, dynamic models of the figure are either constrained to very specific motions such as walking, or have limited predictive ability. However, while the problem domains are different in practice, the approach to singularity analysis is the same in both cases.

6. Experimental Results

We present three sets of experimental results that demonstrate the use of the template plane appearance model for 3D tracking, the use of the SPM model for 2D registration, and a comparison between their performance on a real image sequence. The results include a 3D mouse cursor user-interface which is based on 3D tracking, and a simple example of video-based motion capture using 2D registration.

6.1. 3D Hand Tracking for a 3D Cursor Interface

The first experiment is based on a real-time 3D hand tracking system called *DigitEyes* that was developed by Rehg and Kanade circa 1993 and is described in Rehg (1995) and Rehg

and Kanade (1994b). We apply this system to a user-interface task of tracking the first finger and palm of the hand and using the hand motion to drive the motion of a cursor in a 3D graphical interface (Rehg and Kanade 1994a). This experiment is significant in the context of this paper for two reasons. First, it provides experimental evidence for the utility of the planar template model in hand tracking. Secondly, the experiment demonstrates that when care is taken to avoid visual singularities, good results can be obtained through direct 3D tracking.

6.1.1. A Kinematic Hand Model

We begin by describing the hand model which was employed in the *DigitEyes* system. Kinematic models for visual tracking need only describe motion which a camera can measure. As a result, they can be considerably simpler than those developed by the biomechanics community, which has a long history of skeletal kinematic modeling and analysis. The palm, for example, can be treated as a rigid body even though the four major metacarpal bones lend subtle DOFs.

Attached to the palm are five kinematic chains of three links each, comprising the four fingers and thumb. These chains and their kinematics are illustrated in Figure 8. Each of the four fingers are modeled as planar mechanisms with four DOFs. The abduction DOF moves the plane of the finger relative to the palm, while the remaining three DOFs determine the finger's configuration within the plane. Each finger has an *anchor point*, which is the position of the center of its metacarpophalangeal (MCP) joint in the frame of the palm. Finger chains are built up from revolute joints, as illustrated in the figure. The model for the user-interface experiment involves only the first and fourth finger and the thumb from the full hand model.

We employed the standard DH model to parametrize the kinematic DOFs in the model. The frame for link i is chosen so that the DH parameter θ_i is the revolute joint angle, and the negative x-axis is aligned with $\hat{\mathbf{a}}_i$. With this choice of coordinates, the DH kinematic parameters d_i and α_i are zero, and a_i equals the link length. The complete revolute joint transform is given by

$$\mathbf{T}_i^{i+1} = \mathbf{Rot}_z(\theta_i)\mathbf{Trans}_x(L_i) \quad , \tag{31}$$

where L_i is the length of the ith link. The link lengths are the fixed parameters in the kinematic model. The full kinematic parameters for the hand model are described in the Appendix.

Like the fingers, the thumb model is also constructed from the revolute joints of eq. (31). The thumb is the most difficult digit to model, due to its great dexterity and intricate actuation. It has five DOFs, but one DOF at the trapeziometacarpal joint is dependent on the others. It acts to rotate the thumb longitudinally, bringing it into opposition with the fingers during grasping. Although the visual effect of this rotation is not pronounced, it is included in the current hand model for completeness. This effect is modeled by placing an additional revolute DOF at the thumb MP joint, as shown in Figure 8. Placing the oppositional DOF there, rather than at the base, helps limit its impact on the model.

Our choice of thumb model was motivated by the experience of Rijpkema and Girard (1991) in their grasp modeling system. They employed a similar thumb model and obtained realistic computer graphic animations of hand grasps. Aside from this extra joint, the thumb model is quite similar to that of the fingers, with two DOFs at the trapeziometacarpal joint and one each at the thumb MP and IP joints.

Real hands deviate from the above modeling assumptions in three main ways. First, most fingers are slightly nonplanar. This deviation could be modeled by allowing nonparallel joint axes, but the planar approximation has proved to be adequate in practice. Secondly, the last two joints of the finger (the distal and proximal interphalangeal joints) are driven by the same tendon and are not capable of independent actuation. It is simpler to include these DOFs separately, however, than to model the complicated angular relationship between them. The third deviation stems from the rigid palm assumption, which ignores the metacarpocarpal joints at the base of fingers 4 and 5. When gripping an object, such as a baseball, these joints permit the palm to conform to its surface, causing the anchor points to move by tens of millimeters. For free motions of the hand in space, however, this deviation is small enough to ignore.

Calibration of the fixed parameters of the hand kinematic model followed a two step process. First, link lengths for the first author's hand were measured using a ruler. Secondly, projections of the 3D kinematic model in canonical poses were rendered as on overlay on a live video stream. Alignment of the real hand with the overlaid skeleton revealed errors in the model which were manually adjusted. See Rehg (1995) for a detailed description of this process. The location of the camera relative to the workspace for the user-interface task was determined by off-line calibration, using the procedure described in Robert (1995).

6.1.2. 3D Mouse User-Interface

Hand motion estimated in real time by the *DigitEyes* system using a simplified hand model was employed to drive a 3D mouse interface (Rehg and Kanade 1993, 1994a). Figure 9 shows an example of a simple 3D graphical environment, consisting of a ground plane, a 3D cursor (drawn as a pole, with the cursor at the top), and a spherical object (for manipulation.) Shadows generate additional depth cues. The interface problem is to provide the user with control of the cursor's three DOFs, and thereby the means to manipulate objects in the environment.

In the standard "mouse pole" solution, the 3D cursor position is controlled by clever use of a standard 2D physical mouse. Normal mouse motion controls the base position of the pole on the ground plane. Depressing one of the mouse buttons switches reference planes, causing mouse motion in

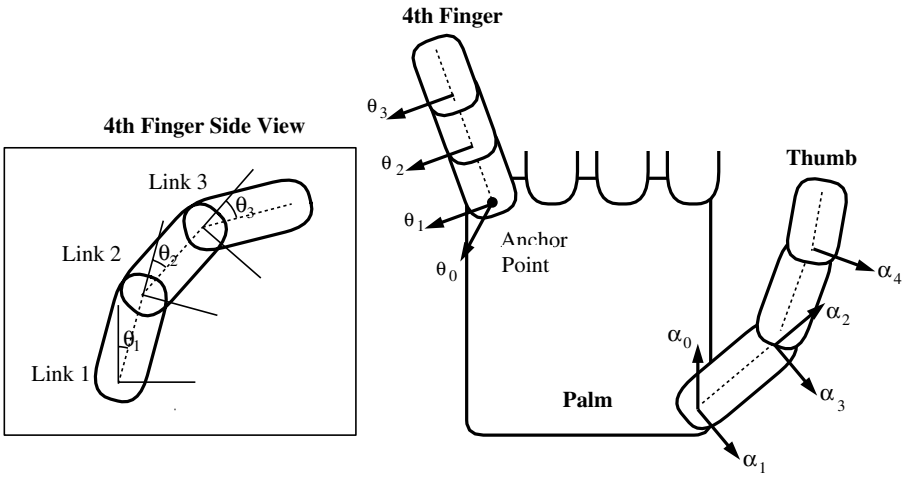

Fig. 8. Kinematic models, illustrated for fourth finger and thumb. The arrows illustrate the joint axes for each link in the chain.

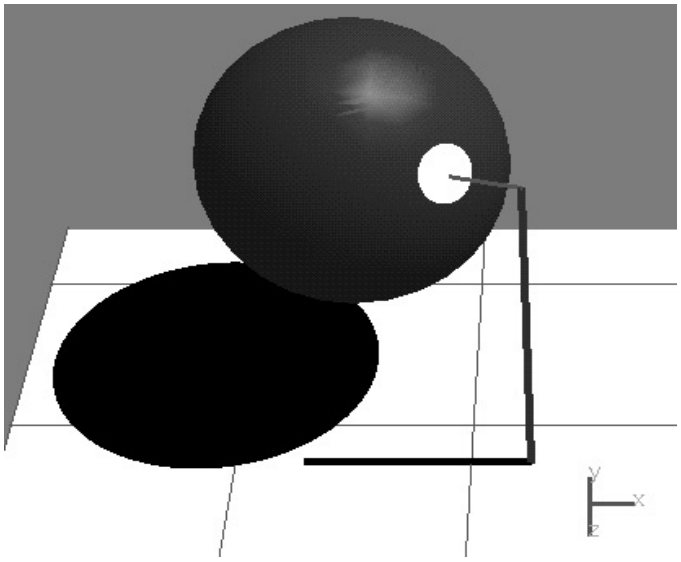

Fig. 9. A sample graphical environment for a 3D mouse. The 3D cursor is at the tip of the "mouse pole", which sits atop the ground plane (in the foreground, at the right). The sphere is an example of an object to be manipulated, and the line drawn from the mouse to the sphere indicates its selection for manipulation.

one direction to control the pole (cursor) height. By switching between planes, the user can place the cursor arbitrarily. Commanding continuous motion with this interface is awkward, however, and tracing an arbitrary, smooth space curve is nearly impossible. *DigitEyes* was used to develop a 3D virtual mouse, that permitted simultaneous hand-based control of the cursor's DOFs.

In the *DigitEyes* solution to the 3D mouse problem, the three input DOFs are derived from a partial hand model, which consists of the first and fourth fingers of the hand, along with the thumb. The palm is constrained to lie in the plane of the table used in the interface, and thus has three DOFs. The first finger has three articulated DOFs, while the fourth finger and thumb each have a single DOF allowing them to rotate in the plan of the table (abduct). The hand model is illustrated in Figure 10. A single camera oriented at approximately 45 degrees to the table top acquires the images used in tracking. The palm position in the plane controls the base position of the pole, while the height of the index finger above the table controls the height of the cursor. This particular mapping has the important advantage of decoupling the controlled DOFs, while making it possible to operate them simultaneously. For example, the user can change the pole height while leaving the base position constant. The fourth finger and thumb have abduction DOFs in the plane, and are used as "buttons".

Figures 11–13 give experimental results from a 500 frame motion sequence in which the estimated hand state was used to drive the 3D mouse interface. Figures 11 and 12 show the estimated hand state for each frame in the image sequence. Frames were acquired at 100 ms sampling intervals. The pole height and base position derived from the hand state by the 3D mouse interface are also depicted in Figure 12. The motion sequence has four phases. In the first phase (frame 0 to 150), the user's finger is raised and lowered twice, producing two peaks in the pole height, with a small variation in the estimated pole position. Secondly, around frame 150 the finger is raised again and kept elevated, while the thumb is actuated, as for a "button event". The actuation period is from frame 150 to frame 200, and results in some change in the pole height, but negligible change in pole position. Thirdly, from frame 200 to 350, the pole height is held constant while the pole position is varied. Finally, from frame 350 to the end of the sequence all states are varied simultaneously. Sample mouse pole positions throughout the sequence are illustrated in Figure 13. This is the same scene as in Figure 9, except that the mouse pole height and position change as a function of the estimated hand state.

6.1.3. Evaluation of 3D Mouse Performance

Following extensive experimentation with the 3D mouse interface, three important attributes of the hand tracker's performance were identified. These factors seemed to have the largest impact on the successful use of the interface:

- sampling rate;
- sensitivity;
- latency.

The quality of the interface as a whole seemed to depend on another set of three properties, which are closely linked to the tracker attributes above:

- maximum hand speed;
- transient DOF coupling;
- resolution.

To illustrate the impact of the tracker's performance on an application, each of the above issues is examined in turn. The maximum possible speed of the user's hand across the table top is a function of the sampling rate of the estimation algorithm, in relation to the error surface properties of the residual. In the specific case of the virtual mouse interface, the tracker could tolerate hand motions of about 2.5 in s^{-1} before track loss began. This was measured experimentally by timing repeated hand translations in the plane, keeping the tracker on the edge of convergence by observing the real-time overlay of the backprojected model and images.

Transient coupling between DOFs is a second factor that is affected by the sampling rate. State coupling is a natural consequence of the kinematic constraints which make tracking possible. These constraints lead to transient effects in the estimator, however, that can negatively impact performance. An example of transient coupling occurs around frame 150 in the button state and mouse pole interface plots from Figures 11 and 12. When the thumb is actuated for a button event, the pole height drops initially, and then rises back to its previous level over the course of about 20 frames. This behavior is the result of an initial tendency of the estimator to spread residual error over all of the states that can reduce it. Only after the thumb has had time to rotate, and absorb most of its residual error, are the other residuals able to reassert their control over their own DOFs. The duration of these transient effects is primarily a result of the sampling rate. More iterations per second make the estimator "stiffer", and reduce the effect of these disturbances.[4] Interestingly, very similar experimental observations have been made in the domain of robot control (Khosla 1986).

The last property of the interface, the resolution with which the cursor position can be controlled, is largely a function of the estimator sensitivity. As described in Section 3, the sensitivity of a state varies with position in the state space. The large-scale effect of this is that the ease of use of the interface depends strongly on the palm orientation relative to the camera. Consider rotating the palm on the table as the pole height

4. Note that the results in this section were obtained from an implementation on a Sun workstation in the early 1990s. An implementation on a modern CPU should yield significant performance increases.

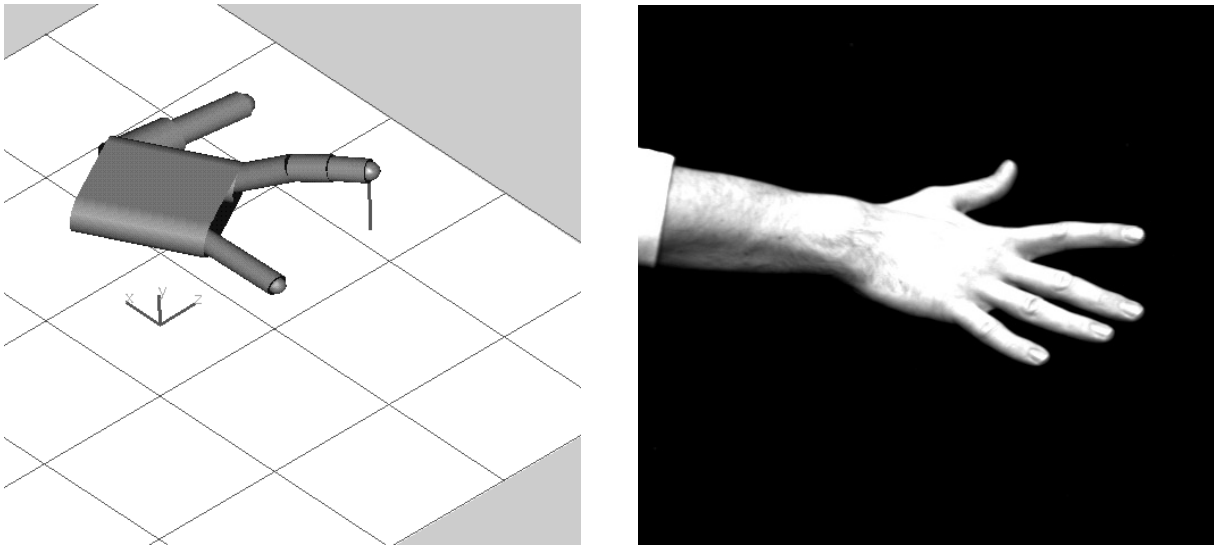

Fig. 10. The hand model used in the 3D mouse application is illustrated for frame 200 in the motion sequence from Figure 12. The vertical line shows the height of the tip above the ground plane. The input hand image (frame 200) demonstrates the finger motion used in extending the cursor height.

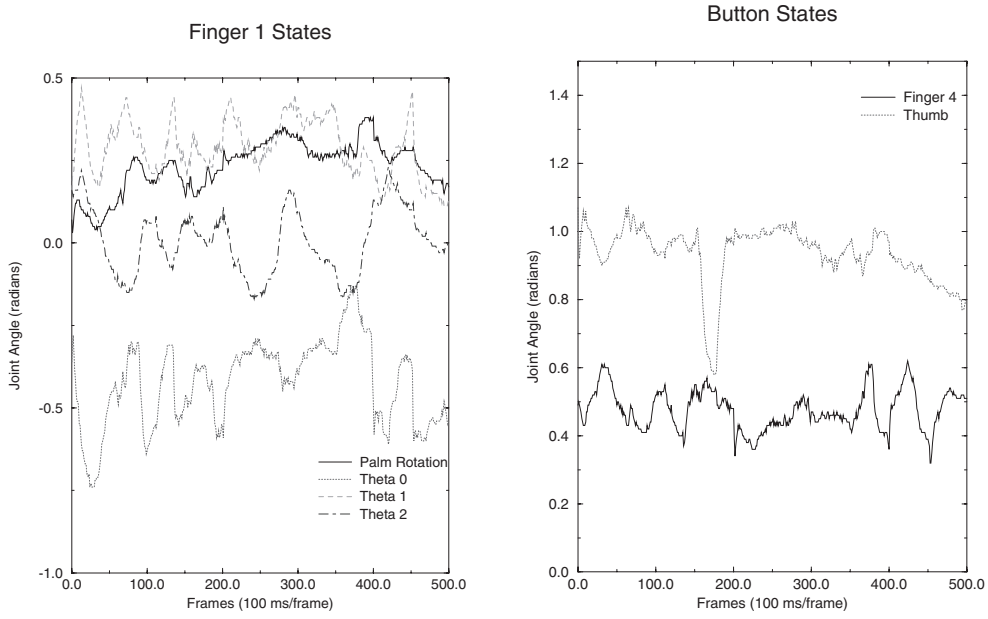

Fig. 11. Palm rotation and finger joint angles for mouse pole hand model depicted in Figure 10. Joint angles for thumb and fourth finger, shown on right, are used as buttons. Note the "button event" signaled by the thumb motion around frame 175.

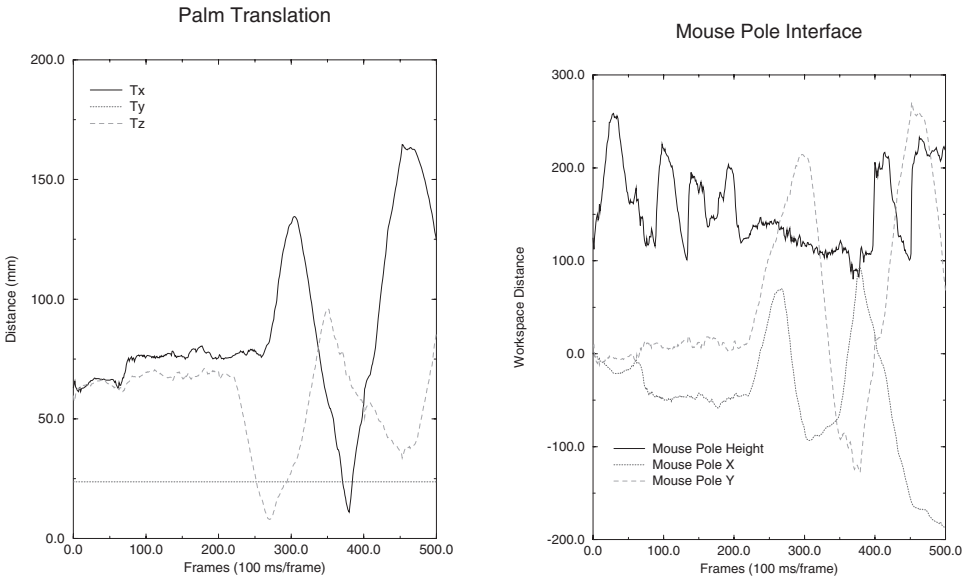

Fig. 12. Translation states for the mouse pole hand model are given on the left. The Y-axis motion is constrained to zero due to the table top. On the right are the mouse pole states, derived from the hand states through scaling and a coordinate change. The sequence events are as follows: 0–150 finger raise/lower; 150–200 thumb actuation only; 200–350 base translation only; 350–500 combined 3-DOF motion.

is varied. The orientation at which the plane of the finger contains the camera is a singular configuration, and pole height becomes extremely difficult to measure. The sensitivity of the estimator to the finger motion decreases as this singularity is approached. The effective resolution in the cursor position is determined by the state sensitivity. The more sensitive the state, the larger the range of image displacements that are produced by a given range of state space motion. This in turn leads to a larger resolution in state space, and greater ability to control the cursor at a fine level of detail.

The effects of latency were not studied in detail for the virtual 3D mouse problem, as they were not extremely significant. Latency refers to the time delay between hand motion and the response of the interface. Long latency times make control of the interface impossible. As a result of the virtual 3D mouse interface design, the total latency was determined by the estimator cycle time, the communication delay to the graphics workstation, and the model rendering time. These last two additional effects added around 30 ms to the 100 ms cycle time. The effect of the total latency was noticeable, but did not make the cursor uncontrollable.

6.2. Comparison between 2D and 3D Models

The second set of experiments compare the performance of 2D and 3D models when tracking through visual singularities. Figures 14(a) and 14(b) show the starting and ending frames of a 30 frame sequence of an arm moving through a singularity. In this example the arm remains rigid, approximating the model of Figure 4(b), but with the addition of a base link capable of translation in the image plane. The trajectory of the arm was similar to the simulation in Figure 5, but with the addition of a non-zero θ component. Overlaid on the images are the positions of the 2D SPM resulting from the state estimates. The longer part of the "T" shape on the arm is the prismatic joint axis. The second link superimposed on the torso has X and Y translation DOFs, which were negligible.

We conducted three experiments in which the sequence in Figure 14 was tracked with an SPM and two 3D kinematic models with different damping factors Λ. In each case, the tracker was given a budget of twenty iterations with which to follow the motion in a given frame. By analogy to the simulation example, we would expect the 3D models to lose ground in the vicinity of a singularity. Figures 15(a)–(c) compare the relative performance of the 2D and 3D models. Figure 15(a) shows the length of the arm link projected into the image plane for the three trackers. As expected, the 2D SPM tracker is unaffected by the singularity and exhibits uniform convergence rates throughout the trajectory. The extension of the arm corresponds to the prismatic state d_1 in the SPM model, which is plotted with large dots in the figure.

The under-damped 3D tracker drawn with dashed lines in Figure 15(a) performs well until it approaches the singularity, upon which it begins oscillating wildly. These oscillations in projected arm length are the result of fluctuations in the out-of-plane rotation angle, which is shown in Figure 15(b). Once the arm leaves the singular configuration, the under-damped tracker recovers and tracks the remainder of the sequence. In contrast, the well-damped tracker plotted with a solid line

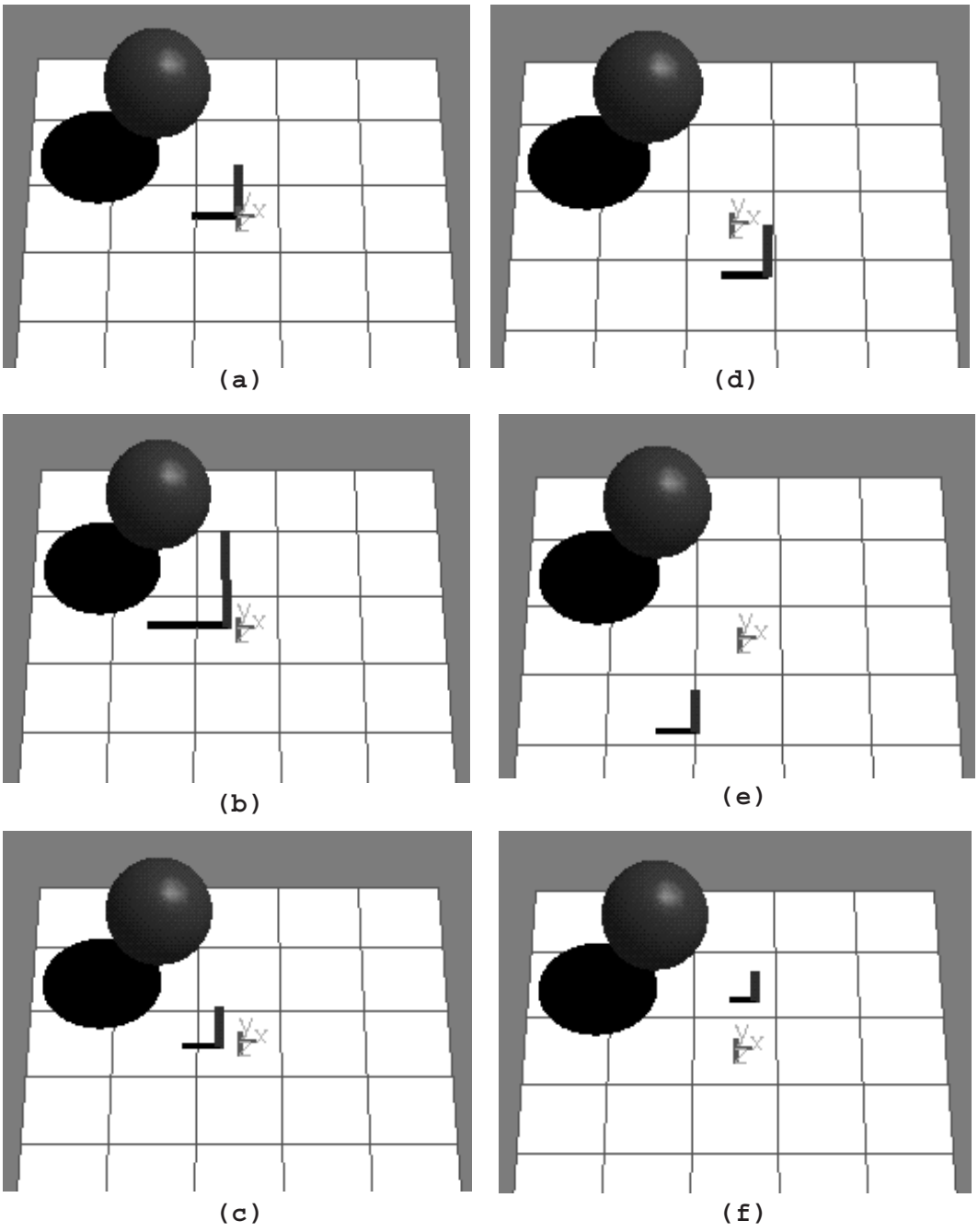

Fig. 13. The mouse pole cursor at six positions during the motion sequence of Figure 11. The pole is the vertical line with a horizontal shadow, and is the only thing moving in the sequence. Samples were taken at frames 0, 30, 75, 260, 300, and 370 (chosen to illustrate the range of motion).

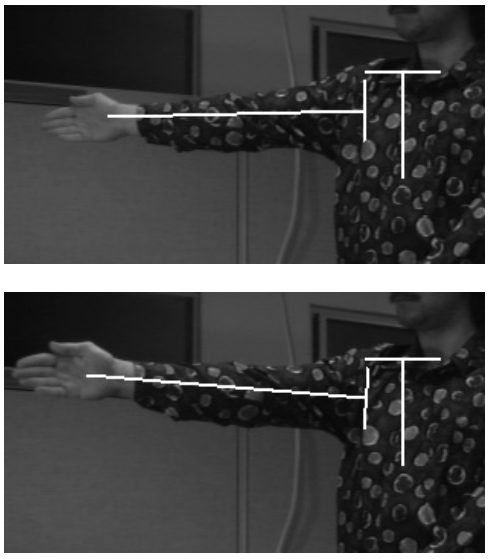

Fig. 14. Test sequence. Frames 15 (a) and 36 (b) from the test sequence for singularity comparison, showing the 2D SPM estimates.

in Figure 15(a) does not oscillate at the singularity. It does, however, have more difficulty escaping from it and lags the SPM tracker by several pixels over several frames of measurements.

In a real application, an algorithm such as the Levenberg–Marquardt algorithm would be used to automatically adapt the amount of damping. It is clear, however, that any 3D tracker will be forced to do a significant amount of work in the vicinity of the singularity to avoid poor performance. Unfortunately, in spite of this effort the 3D tracker will be quite sensitive to both image noise and errors in the kinematic model parameters, such as link lengths, during this part of the trajectory.

Figure 15(b) shows the out-of-plane rotation angle, ϕ, for the two 3D models. The divergence of the two curves following the singularity is a consequence of the usual orthographic ambiguity. Figure 15(c) shows the in-plane rotation angle, θ, which is essentially the same for all of the models. In summary, the 2D SPM exhibits more consistent and uniform registration performance, as expected. The performance of the 3D model depends critically on determining the correct amount of damping.

6.3. Motion Capture from Movies

For the third experiment, we developed a 2D SPM for the human figure and applied it to a short dance sequence by Fred Astaire. Figure 16 shows stills from the movie "Shall We Dance", overlaid with their associated state estimates. The tracker was initialized on the first frame of the video sequence by manually aligning a skeletal SPM with the torso, limbs, and

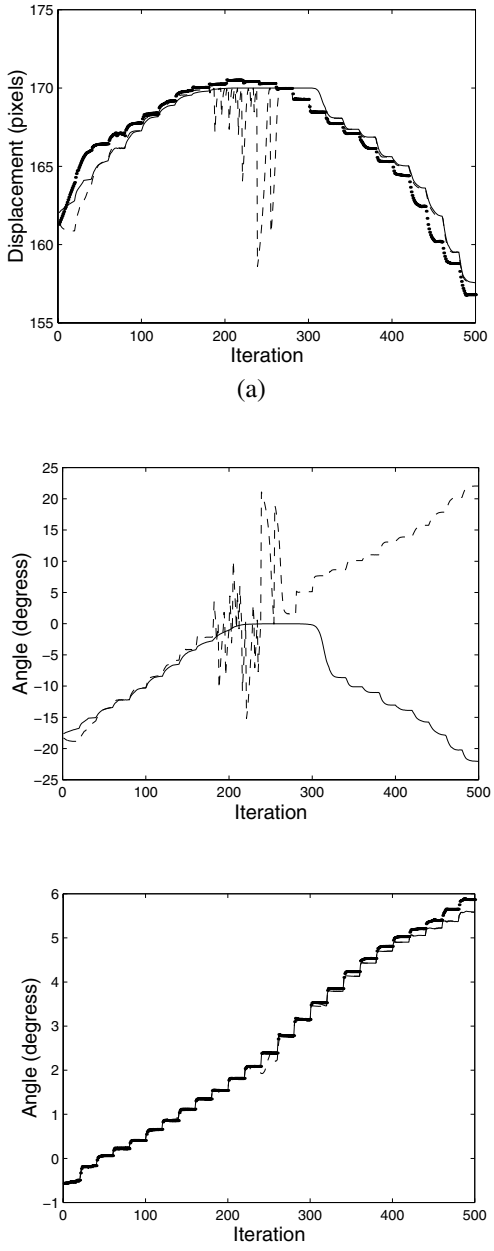

Fig. 15. Tracking results for 2D SPM and 3D kinematic models using the motion sequence in Figure 14. 2D SPM data are shown by large dots, while 3D model data are shown by a solid curve in the well-damped case and a dashed line in the under-damped case. (a) (Top) Displacement in pixels corresponding to the length of the arm link after projection into image plane using the estimated state. (b) (Middle) Angle ϕ of 3D trackers. (c) (Bottom) In-plane rotation θ of each model.

Fig. 16. Fred Astaire tracked in an image sequence using the SPM-based tracker. Images (a)–(f) correspond to frames 1, 5, 9, 11, 15, and 19 from the input sequence.

Table 1. DH Kinematic Model for the First Author's Right Hand

Frame	Geometry	θ	d	a	α	Shape (in mm)	Next
0	Palm	0.0	0.0	0.0	0.0	x 56.0, y 86.0, z 15.0	1 8 15 22 29
1		$\pi/2$	0.0	38.0	$-\pi/2$		2
2		0.0	−31.0	0.0	$\pi/2$		3
3		$\mathbf{q}_7$	0.0	0.0	$\pi/2$		4
4	Finger 1 Link 0	$\mathbf{q}_8$	0.0	45.0	0.0	Rad 10.0	5
5	Finger 1 Link 1	$\mathbf{q}_9$	0.0	26.0	0.0	Rad 10.0	6
6	Finger 1 Link 2	$\mathbf{q}_{10}$	0.0	24.0	0.0	Rad 9.0	7
7	Finger 1 Tip	0.0	0.0	0.0	0.0	Rad 9.0	–
8		$\pi/2$	0.0	37.0	$-\pi/2$		9
9		0.0	-9.0	0.0	$\pi/2$		10
10		$\mathbf{q}_{11}$	0.0	0.0	$\pi/2$		11
11	Finger 2 Link 0	$\mathbf{q}_{12}$	0.0	56.0	0.0	Rad 10.0	12
12	Finger 2 Link 1	$\mathbf{q}_{13}$	0.0	27.0	0.0	Rad 10.0	13
13	Finger 2 Link 2	$\mathbf{q}_{14}$	0.0	22.0	0.0	Rad 9.0	14
14	Finger 2 Tip	0.0	0.0	0.0	0.0	Rad 7.0	–
15		$\pi/2$	0.0	33.0	$-\pi/2$		16
16		0.0	6.0	0.0	$\pi/2$		17
17		$\mathbf{q}_{15}$	0.0	0.0	$\pi/2$		18
18	Finger 3 Link 0	$\mathbf{q}_{16}$	0.0	53.0	0.0	Rad 9.0	19
19	Finger 3 Link 1	$\mathbf{q}_{17}$	0.0	25.0	0.0	Rad 9.0	20
20	Finger 3 Link 2	$\mathbf{q}_{18}$	0.0	20.0	0.0	Rad 8.0	21
21	Finger 3 Tip	0.0	0.0	0.0	0.0	Rad 7.0	–
22		$\pi/2$	0.0	30.0	$-\pi/2$		23
23		0.0	26.0	0.0	$\pi/2$		24
24		$\mathbf{q}_{19}$	0.0	0.0	$\pi/2$		25
25	Finger 4 Link 0	$\mathbf{q}_{20}$	0.0	38.0	0.0	Rad 9.0	26
26	Finger 4 Link 1	$\mathbf{q}_{21}$	0.0	19.0	0.0	Rad 8.0	27
27	Finger 4 Link 2	$\mathbf{q}_{22}$	0.0	17.0	0.0	Rad 7.0	28
28	Finger 4 Tip	0.0	0.0	0.0	0.0	Rad 6.0	–
29		$-\pi/2$	15.0	43.0	$-\pi/2$		30
30		$-\pi$	38.0	0.0	0.0		31
31		$\mathbf{q}_{23}$	0.0	0.0	$\pi/2$		32
32	Thumb Link 0	$\mathbf{q}_{24}$	0.0	46.0	$-\pi/2$	Rad 14.0	33
33		$\mathbf{q}_{25}$	0.0	0.0	$\pi/2$		34
34	Thumb Link 1	$\mathbf{q}_{25}$	0.0	34.0	0.0	Rad 10.0	35
35	Thumb Link 2	$\mathbf{q}_{26}$	0.0	25.0	0.0	Rad 10.0	36
36	Thumb Tip	0.0	0.0	0.0	0.0	Rad 8.0	–

head of Fred Astaire. A fixed width for each of the link templates was determined manually. Templates models for each link were acquired from the first frame and used throughout the sequence. We used the standard SSD likelihood model in which the probability of a particular configuration of the SPM model is determined by the squared pixel error between the set of templates and the image. This objective function was minimized using standard Levenberg–Marquardt techniques. As a preprocessing step, we used standard image stabilization techniques (Irani, Rousso, and Peleg 1994) to compensate for the camera motion so that the figure is the only moving object in the scene.

The overall quality of the registration in Figure 16 is good, especially considering the low contrast between figure and background. Frames (b) and (c) exhibit the most artifacts, some of which could be fixed by integration with a 3D kinematic model. Note that while probabilistic tracking techniques could possibly improve the tracking performance, our goal here is simply to demonstrate the ease with which the SPM model can be applied to a complex motion sequence.

7. Conclusions

While kinematic models provide powerful constraints for gradient-based tracking algorithms, we have shown that trackers utilizing 3D kinematic models suffer from singularities when motion is directed along the viewing axis of a single camera. This results in greater sensitivity to noise and possible loss of registration.

We have introduced a 2D SPM which captures the image plane motion of a large class of 3D kinematic models. The SPM has the following three advantages. First, it has fewer singularity problems than 3D kinematic models. In addition, unlike the general 3D model, its singularities can be fully characterized enabling it to be used only in appropriate situations. Secondly, the SPM does not require the specification of link lengths and joint axes, which can sometimes be difficult. In cases where 3D information is unnecessary, the SPM alone may provide sufficient motion estimation. Thirdly, when 3D motion estimates are desired, they can be obtained from SPM motion estimates using a batch estimation approach.

We have proposed an alternative to the direct 3D tracking approach, based on decomposing figure tracking into separate *image registration* and *3D reconstruction* stages. This is consistent with standard approaches to structure from motion problems (Tomasi and Kanade 1992). This decomposition has two potential benefits. First, the registration stage can employ simple 2D figure models which avoid most of the singularity problems associated with 3D tracking in the case of a single video source. Secondly, the reconstruction stage can simultaneously estimate both dynamic state parameters, such as joint angles, and static parameters, such as link lengths. This would remove the need to specify an accurate figure model for 3D tracking.

We have demonstrated direct 3D tracking results for a mouse cursor user-interface. This result demonstrates the utility of the 3D kinematic constraint when visual singularities can be avoided. We have also used the SPM to track Fred Astaire in a video clip taken from the movie "Shall We Dance".

Appendix: Whole Hand DH Model

Table 1 gives the full DH model for the first author's right hand, which was used in the mouse pole user-interface experiment.

References

Bergen, J., et al. 1992. Hierarchical model-based motion estimation. In *Second European Conference on Computer Vision*, pp. 237–252, Santa Margherita Liguere, Italy, Springer-Verlag, Berlin.

Brand, M. 1999. Shadow puppetry. In *Proceedings of International Conference on Computer Vision*, Vol. II, pp. 1237–1244, Kerkyra, Greece.

Bregler, C., and Malik, J. 1998. Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition*, pp. 8–15, Santa Barbara, CA.

Cham, T.-J., and Rehg, J. M. 1999. A multiple hypothesis approach to figure tracking. In *Computer Vision and Pattern Recognition*, Vol. II, pp. 239–245, Fort Collins, Colorado.

Cox, I. J., and Hingorani, S. L. 1996. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(2):138–150.

Delamarre, Q., and Faugeras, O. 2001. 3D articulated models and multi-viewtracking with physical forces. *Computer Vision and Image Understanding (CVIU)*, 81:328–357.

Dennis, J., and Schnabel, R. 1983. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.

Deutscher, J., Blake, A., and Reid, I. 2000. Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition*, Vol. 2, pp. 126–133, Hilton Head, SC.

DiFranco, D. E., Cham, T.-J., and Rehg, J. M. November 2001. Reconstruction of 3D figure motion from 2D correspondences. In *Computer Vision and Pattern Recognition*, Kauai, Hawaii.

Espiau, B., Chaumette, F., and Rives, P. 1992. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8:313–326.

Fablet, R., and Black, M. J. 2002. Automatic detection and tracking of human motion with a view-based representation. In *European Conference on Computer Vision (ECCV)*, pp. 476–491.

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

Goncalves, L., Bernardo, E. D., Ursella, E., and Perona, P. 20–23 June 1995. Monocular tracking of the human arm in 3D. In *Proceedings of International Conference on Computer Vision*, pp. 764–770, Cambridge, MA.

Heap, A. J., and Hogg, D. C. 1998. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Proceedings of International Conference on Computer Vision*, Bombay, India.

Hogg, D. 1983. Model-based vision: a program to see a walking person. *Image Vision Computing* 1(1):5–20.

Howe, N., Leventon, M., and Freeman, W. November 1999. Bayesian reconstruction of 3D human motion from single-camera video. In *Neural Information Processing Systems (NIPS)*, Denver, CO.

Irani, M., Rousso, B., and Peleg, S. 1994. Computing occluding and transparent motions. *International Journal of Computer Vision* 12(1):5–16.

Isard, M., and Blake, A. April 1996. Contour tracking by

stochastic propagation of conditional density. In *European Conference on Computer Vision (ECCV)*, Vol. I, pp. 343–356, Cambridge, UK.

Ju, S. X., Black, M. J., and Yacoob, Y. 1996. Cardboard people: A parameterized model of articulated image motion. In *International Conference of Automatic Face and Gesture Recognition*, pp. 38–44, Killington, VT.

Kakadiaris, I., and Metaxas, D. June 1995. 3D human body model acquisition from multiple views. In *Proceedings of International Conference on Computer Vision*, pp. 618–623, Cambridge, MA.

Kakadiaris, I., and Metaxas, D. 18–20 June 1996. Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. In *Computer Vision and Pattern Recognition*, pp. 81–87, San Francisco, CA.

Kanatani, K., and Morris, D. D. 2001. Gauge and gauge transformations for uncertainty description of geometric structure with indeterminacy. *IEEE Transactions on Information Theory* 47(5):2017–2028.

Khosla, P. 1986. Real-time Control and Identification of Direct-Drive Manipulators. PhD Thesis, Carnegie Mellon University, Department of ECE.

Koenderink, J. J., and van Doorn, A. J. 1991. Affine structure from motion. *Journal of the Optical Society of America* 8(2):377–385.

MacCormick, J., and Blake, A. 1999. A probabilistic exclusion principle for tracking multiple objects. In *Proceedings of International Conference on Computer Vision*, pp. 572–578.

MacCormick, J., and Isard, M. 2000. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *European Conference on Computer Vision (ECCV)*, Vol. 2, pp. 3–19.

McCarthy, J. 1990. *An Introduction to Theoretical Kinematics*, MIT Press, Cambridge, MA.

Morris, D. D., and Rehg, J. M. 23–25 June 1998. Singularity analysis for articulated object tracking. In *Computer Vision and Pattern Recognition*, pp. 289–296, Santa Barbara, CA.

Nakamura, Y. 1991. *Advanced Robotics: Redundancy and Optimization*, Addison-Wesley, Reading, MA.

Nelson, B. J., and Khosla, P. K. 1994. The resolvability ellipsoid for visual servoing. In *Computer Vision and Pattern Recognition*, pp. 829–832.

O'Rourke, J., and Badler, N. 1980. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2(6):522–536.

Pai, D. K. May 1988. Singularity, Uncertainty, and Compliance of Robot Manipulators. PhD Thesis, Cornell University, Ithaca, NY.

Papanikolopoulos, N., Khosla, P., and Kanade, T. 1993. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Transactions on Robotics and Automation* 9(1):14–35.

Pavlović, V., Rehg, J., Cham, T., and Murphy, K. 1999. A dynamic Bayesian network approach to figure tracking using learned dynamic models. In *Proceedings of International Conference on Computer Vision*, Vol. I, pp. 94–101, Corfu, Greece.

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

Rehg, J. M. April 1995. Visual Analysis of High DOF Articulated Objects with Application to Hand Tracking. PhD Thesis, Carnegie Mellon University. Available as School of Computer Science Technical Report CMU-CS-95-138.

Rehg, J. M. January 2000. Motion capture from movies. In *Proceedings of Asian Conference on Computer Vision*, Vol. II, pp. 1125–1131, Taipei, Taiwan. Invited Paper.

Rehg, J. M., and Kanade, T. 1993. Digiteyes: Vision-based human hand tracking. Technical Report CMU-CS-TR-93-220, Carnegie Mellon University, School of Computer Science.

Rehg, J. M., and Kanade, T. 1994a. Digiteyes: Vision-based hand tracking for human-computer interaction. In J. K. Aggarwal and T. S. Huang, editors, *Proceedings of Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 16–22, Austin, TX.

Rehg, J. M., and Kanade, T. 1994b. Visual tracking of high DOF articulated structures: an application to human hand tracking. In *European Conference on Computer Vision (ECCV)*, Vol. 2, pp. 35–46, Stockholm, Sweden.

Rehg, J. M., and Kanade, T. 1995. Model-based tracking of self-occluding articulated objects. In *Proceedings of International Conference on Computer Vision*, pp. 612–617, Cambridge, MA.

Rehg, J. M., Kang, S. B., and Cham, T.-J. September 2000. Video editing using figure tracking and image-based rendering. In *International Conference on Image Processing*, Vancouver, B.C. Invited Paper.

Rehg, J. M., and Morris, D. D. 1997. Singularities in articulated object tracking with 2D and 3D models. Technical Report CRL 97/8, DEC Cambridge Research Lab.

Rijpkema, H., and Girard, M. 1991. Computer animation of knowledge-based human grasping. *Computer Graphics* 25(4):339–348.

Robert, L. 1995. Camera calibration without feature extraction. *Computer Vision, Graphics, and Image Processing* 63(2):314–325.

Rohr, K. 1994. Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics, and Image Processing* 59(1):94–115.

Rowley, H. A., and Rehg, J. M. 17–19 June 1997. Analyzing articulated motion using expectation-maximization. In *Computer Vision and Pattern Recognition*, pp. 935–941, San Juan, Puerto Rico.

Sharma, R., and Hutchinson, S. August 1997. Motion per-

ceptibility and its application to active vision-based servo control. *IEEE Transactions on Robotics and Automation* 13(4):607–617.

Shimada, N., Shirai, Y., Kuno, Y., and Miura, J. 1998. Hand gesture estimation and model refinement using monocular camera—ambiguity limitation by inequality constraints. In *Proceeding of the Third International Conference on Automatic Face and Gesture Recognition*, pp. 268–273, Nara, Japan.

Sidenbladh, H., Black, M. J., and Fleet, D. J. 2000. Stochastic tracking of 3D human figures using 2D image motion. In *Proceedings of the European Conference on Computer Vision*, Dublin, Ireland.

Sidenbladh, H., De la Torre, F., and Black, M. J. March 2000. A framework for modeling the appearance of 3D articulated figures. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 368–375, Grenoble, France.

Sminchisescu, C., and Triggs, B. 2001. Covariance scaled sampling for monocular 3D body tracking. In *Computer Vision and Pattern Recognition*, Vol. 1, pp. 447–454.

Spong, M. 1989. *Robot Dynamics and Control*, Wiley, New York.

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

Toyama, K., and Blake, A. 2001. Probabilistic tracking in a metric space. In *Proceedings of International Conference on Computer Vision*, Vancouver, Canada.

Wachter, S., and Nagel, H.-H. June 1999. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding (CVIU)* 74(3):174–192.

Weiss, L., Sanderson, A., and Neuman, C. October 1987. Dynamic sensor-based control of robots with visual feedback. *IEEE Transactions on Robotics and Automation*, 3(5):404–417.

Yamamoto, M., and Koshikawa, K. 1991. Human motion analysis based on a robot arm model. In *Computer Vision and Pattern Recognition*, pp. 664–665. Also see ETL TR 90-46.