Int J Comput Vis (2008) 79: 247–269
DOI 10.1007/s11263-007-0116-2

Human Motion Tracking with a Kinematic Parameterization of Extremal Contours

David Knossow · Rémi Ronfard · Radu Horaud

Received: 26 July 2006 / Accepted: 19 November 2007 / Published online: 8 December 2007

Abstract This paper addresses the problem of human motion tracking from multiple image sequences. The human body is described by five articulated mechanical chains and human body-parts are described by volumetric primitives with curved surfaces. If such a surface is observed with a camera, an extremal contour appears in the image whenever the surface turns smoothly away from the viewer. We describe a method that recovers human motion through a kinematic parameterization of these extremal contours. The method exploits the fact that the observed image motion of these contours is a function of both the rigid displacement of the surface and of the relative position and orientation between the viewer and the curved surface. First, we describe a parameterization of an extremal-contour point velocity for the case of developable surfaces. Second, we use the zero-reference kinematic representation and we derive an explicit formula that links extremal contour velocities to the angular velocities associated with the kinematic model. Third, we show how the chamfer-distance may be used to measure the discrepancy between predicted extremal contours and observed image contours; moreover we show how the chamfer distance can be used as a differentiable multi-valued function and how the tracker based on this distance can be cast into a continuous non-linear optimization framework. Fourth, we describe implementation issues associated with a practical human-body tracker that may use an arbitrary number of cameras. One great methodological and practical advantage of our method is that it relies neither on model-to-image, nor on image-to-image point matches. In practice we model people with 5 kinematic chains, 19 volumetric primitives, and 54 degrees of freedom; We observe silhouettes in images gathered with several synchronized and calibrated cameras. The tracker has been successfully applied to several complex motions gathered at 30 frames/second.

Keywords Articulated motion representation · Human-body tracking · Zero-reference kinematics · Developable surfaces · Extremal contours · Chamfer distance · Chamfer matching · Multiple-camera motion capture

D. Knossow · R. Ronfard · R. Horaud (✉)
INRIA Rhône-Alpes, 655, avenue de l'Europe,
38330 Montbonnot Saint-Martin, France
e-mail: radu.horaud@inrialpes.fr

1 Introduction and Background

In this paper we address the problem of tracking complex articulated motions from multiple image sequences. The problem of articulated motion (such as human-body motion) representation and tracking from 2-D and 3-D visual data has been thoroughly addressed in the recent past. The problem is difficult because it needs to solve an inverse kinematic problem, namely the problem of finding the parameters characterizing the control space (the space spanned by the articulated parameters) from a set of measurements performed in the observation space. In general this problem cannot be solved explicitly because the dimensionality of the observation space is much smaller than the dimensionality of the control space. More formally, the problem can be stated as the following minimization problem:

$$\min_{\mathbf{\Phi}} E(\mathcal{Y}, \mathcal{X}(\mathbf{\Phi})) \tag{1}$$

where $\mathcal{Y}$ denotes a set of observations, $\mathcal{X}$ denotes a set of predictions using the direct kinematic model, and $\mathbf{\Phi}$ is the vector of motion parameters to be estimated.

In this paper we will embed the human-motion tracking into the minimization problem defined by (1). We will emphasize a human-body model composed of articulated mechanical chains and of rigid body parts. Each such part is defined by a developable surface. An intrinsic property of such a surface is that it projects onto an image as a pair of straight extremal contours (an extremal contour appears in an image whenever a curved surface turns smoothly away from the viewer). We develop a direct kinematic representation of extremal contours based on the differential properties of developable surfaces and on the zero-reference kinematic representation of articulated chains with rotational joints. This kinematic description encapsulates the constrained articulated motions as well as a free rigid motion and allows us to predict both the position and the velocity of extremal contours.

Therefore, human motion tracking may be formulated as the problem of minimizing equation (1) using the chamfer distance between predicted extremal contour points, $\mathcal{X}(\mathbf{\Phi})$ and contour points detected in images, $\mathcal{Y}$. We show how the chamfer distance can be used as a differentiable multi-valued function and how the tracker based on this distance can be cast into a non-linear optimization framework. Even if, in theory, one camera may be sufficient for recovering the motion parameters, we show that a multiple-camera setup brings in the necessary robustness for implementing the tracker.

There is a substantial body of computer vision literature on articulated motion tracking and excellent reviews can be found in (Gavrila 1999), (Moeslund et al. 2006), and (Forsyth et al. 2006).

Monocular approaches generally require a probabilistic framework such as in (Deutscher et al. 2000; Toyama and Blake 2002; Song et al. 2003; Agarwal and Triggs 2006) to cite just a few. The probabilistic formulation has the attraction that both prior knowledge and uncertainty in the data are handled in a systematic way. The first difficulty with these methods is that the image data must be mapped onto a vector space with fixed dimension such that statistical methods can be easily applied. The second difficulty is to establish a relationship (between the space of articulated poses and the space spanned by the vectors mentioned above) that should be learnt prior to tracking. This is not an obvious task because it is virtually impossible to scan in advance the space of all possible poses of an articulated object with many degrees of freedom. Other methods attempted to recover articulated motion from image cues such as optical flow through sophisticated non-linear minimization methods (Bregler et al. 2004; Sminchisescu and Triggs 2003; Sminchisescu and Triggs 2005).

A second class of approaches relies on multiple-video sequences gathered with multiple cameras. One pre-requisite of such a camera setup is that the frames are finely synchronized—a not so obvious task. One can either perform some kind of 3-D surface or volumetric reconstruction prior to tracking (Cheung et al. 2005a; Mikic et al. 2003; Plaenkers and Fua 2003), or use 2-D features such as silhouettes, color, or texture (Delamarre and Faugeras 2001; Drummond and Cipolla 2001; Gavrila and Davis 1996; Kakadiaris and Metaxas 2000). Others used a combination of both 2-D and 3-D features (Plaenkers and Fua 2003; Kehl and Van Gool 2006).

In (Cheung et al. 2005a) and (Cheung et al. 2005b) the authors develop a shape-from-silhouette paradigm that is applied to human motion tracking. They describe a volumetric-based method that assigns a voxel to a body part and a method based on colored surface points (CSP) that combines silhouette-based reconstruction with color information. Both these methods require 3-D reconstruction from perfect silhouettes. A similar voxel-based method is described in (Mikic et al. 2003). The tracker minimizes a cost function that measures the consistency between the 3-D data (a set of voxels) and the model (ellipsoids linked within an articulated chain).

In (Kehl and Van Gool 2006) image edges, color, and a volumetric reconstruction are combined to take advantage of these various 2-D and 3-D cues. The authors notice that while volumetric data are strong features, image edges are needed for fine localization and hence accurate pose computation. The use of edges implies that one is able to predict model edges. Since the authors use superquadrics, it is necessary to compute their contour generator (referred in Kehl and Van Gool 2006 as the occluding contour) and project it in the images using perspective projection. This is done through a series of approximations since a closed-form solution is difficult to compute. Finally the authors cast the tracking problem into a stochastic optimization framework that uses a three-term cost function for surface, edge, and color alignment. The experimental setup uses 16 cameras. A similar approach based on both 3-D data (depth from a stereo image pair) and 2-D silhouettes is proposed in (Plaenkers and Fua 2003).

In (Kakadiaris and Metaxas 2000) and (Delamarre and Faugeras 2001) two similar methods are presented. Multiple-camera tracking is performed by projecting the 3-D model onto the images and building a cost function that measure the distance between the projected model and the 2-D silhouettes. This distance sums up the squares of the projected-model-point-to-silhouette-point assignments to estimate the 2-D force field and to infer the "physical forces" that allow the alignment.

In this paper we use neither color nor photometric information because it is not robust to illumination changes. We do not use texture because it is not a shape-invariant feature. We decided to concentrate on contours because they have been recognized as strong cues for representing shape (Koenderink 1990; Forsyth and Ponce 2003) and therefore

the tracker that we implemented projects predicted model contours onto the images and compares them with observed contours (edges, silhouettes, etc.). Nevertheless, the tasks of computing contours from 3-D models, of projecting these contours onto images, and of comparing them with observed ones are not straightforward. Previous methods have not made explicit the analytic representation allowing the mapping of articulated objects (and their surfaces) onto 2-D edges or silhouettes. Formally, a silhouette is the *occluding contour* (Barrow and Tenenbaum 1981) that separates an object from the background. Occluding contours are built up of *discontinuity* and *extremal* contours. The former correspond to sharp edges arising from surface discontinuities. The latter occur where a curved surface turns smoothly away from the viewer.

In the case of sharp edges there are well documented methods allowing for an explicit (analytic) representation of the mapping between the object's constrained (articulated) motion parameters and the observed image contours both under orthography (Bregler et al. 2004) and under perspective projection (Drummond and Cipolla 2001; Martin and Horaud 2002). In the presence of smooth surfaces, an extremal contour is the projection of a *contour generator*—a virtual contour that lies onto the surface where the lines of sight are tangent to the surface. Therefore, the apparent image motion of an extremal contour is a function of both the motion of the object itself and the motion of the contour generator, the latter being a function of the relative position of the object's surface with respect to the viewer. It turns out that the link between the differential properties of certain classes of surfaces and the rigid motion of these surfaces has barely been addressed.

In more detail, we use *elliptical cones* to model body parts. These shapes belong to a more general class of developable surfaces that have interesting differential properties that were not fully exploited in the past. Elliptical cones in particular and developable surfaces in general project onto images as a set of straight lines. By deliberately considering only these contours we simplify both the tasks of interpreting the image contours and of comparing them to the predicted object contours. Moreover, the body parts are joined together to form an articulated structure composed of five *open kinematic chains*. Therefore, each body-part motion is composed of two motions: a motion constrained by a number of rotational joints (the motion of its associated kinematic chain) and a free motion, i.e., the motion of the root body-part with respect to a world reference frame. We derive an analytic expression for the motion of a predicted extremal-contour point as a function of both the body-part motion as well as the motion of its contour generator lying onto the curved surface of that body part. Figure 1 briefly illustrates how the method operates in practice.

Therefore, the problem of articulated motion tracking may be formulated as the problem of minimizing a metric between image contours (gathered simultaneously with several cameras) and extremal contours (predicted from the model). There are several ways of defining a distance between two contours, including the sum of squares of the point-to-point distances, the Hausdorff distance, the chamfer distance, and so forth. We decided to capitalize onto the chamfer distance, and unlike previous approaches, we developed an analytic expression allowing us to compare the real-valued contour points predicted from the model with the chamfer-distance image computed from binary-valued image contours. This image-to-model metric thus defined does not require point-to-point matches, its computation is very efficient, and it can be analytically differentiated. We analyse in detail the numerical conditioning of the tracker, which amounts to the rank analysis of the Jacobian associated with the direct kinematic model. Although, in principle, one camera may be sufficient for gathering enough data, we claim that a multiple-camera setup provides the redundancy that is absolutely necessary for robust tracking.

Paper Organization The remainder of this paper is organized as follows. In Sect. 2 we consider the case of developable surfaces and we show that their contour generators are rulings of the surface. We derive a closed-form solution for the velocity of the contour generators (and of the corresponding extremal contours) as a function of the kinematic screw associated with the motion of the surface. In Sect. 3 we develop an explicit solution for the human-body kinematics using the zero-reference kinematic model and in Sect. 4 we derive the Jacobian that maps joint and free-motion velocities onto the 2-D velocity of an extremal-contour point. Section 5 describes in detail how to fit predicted extremal contours to detected image contours and how to carry out the minimization process using the chamfer distance. Section 6 describes experiments performed with simulated and real data. Finally, Sect. 7 draws some conclusions and give directions for future work.

2 The Kinematics of Extremal Contours

2.1 Definitions and Notations

We use shapes with smooth surfaces in order to represent rigid body parts. Each such body-part is linked to a *root* body-part through a kinematic chain of body parts. Each joint in the kinematic chain—the link between two adjacent body parts—is modeled by a rotational joint. Each such joint may have one, two, or three degrees of freedom. Moreover, the root body-part is allowed to freely move in the 3-D space with six degrees of freedom (three rotations and three translations).

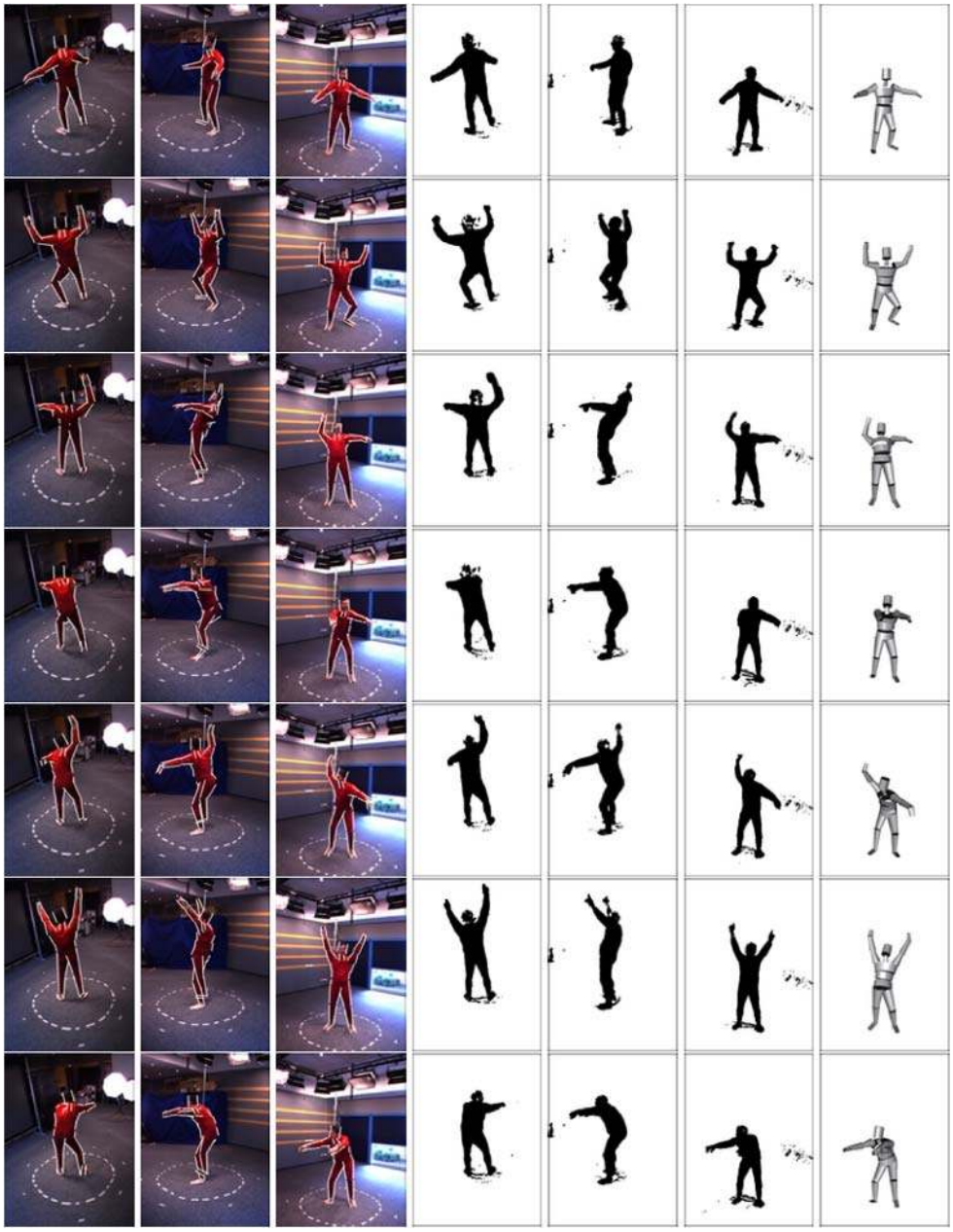


Fig. 1 An example of human-motion tracking based on extremal contours and using six cameras. The extremal contours fitted to the image data are shown superimposed onto the raw images. The tracker uses image silhouettes to fit the parameterized extremal contours to the data. The recovered pose of the human-body model is shown from the viewpoint of the third camera. There are 250 frames in these six image sequence. Notice that this apparent simple gesture (raising the arms and then leaning forward) involves almost all the degrees of freedom of the model as well as a motion of the root body-part

Therefore, the motion of any part of the kinematic chain is obtained by a combination of a *constrained motion* and of a *free motion*. We denote by $\mathbf{\Phi} = (\phi_1, \ldots, \phi_n)$ all these motion parameters. The first q parameters correspond to the motion of the root with $q \leq 6$ and the remaining p parameters correspond to the joint angles: $n = q + p$. The kinematic parameterization will be made explicit in the next section. In this section we will describe the motion of a body-part by a 3×3 rotation matrix $\mathbf{R}$ and by a 3-D translation vector $\boldsymbol{t}$. Both these rotation and translation are in turn parameterized by $\mathbf{\Phi}$, i.e., we will have $\mathbf{R}(\mathbf{\Phi})$ and $\boldsymbol{t}(\mathbf{\Phi})$.

It will also be convenient to consider a body part as a rigid object in its own right. The *pose of a rigid object* is described by six parameters and let $\boldsymbol{r}$ be the pose vector. If a body-part is treated as a free-moving rigid body, then the 6 components of $\boldsymbol{r}$ are the free parameters. If a body-part is treated as a

component of a kinematic chain, $\boldsymbol{r}$ is parameterized by $\boldsymbol{\Phi}$, i.e., $\boldsymbol{r}(\boldsymbol{\Phi})$. Finally we denote by $\dot{\boldsymbol{x}}$ the time derivative of $\boldsymbol{x}$.

We consider now the smooth surface of a body-part. This surface projects onto the image as an extremal contour. The apparent image motion of such an extremal contour depends on the motion of the body-part and on the local shape of the surface. Indeed, let's consider the *contour generator* that lies onto the smooth surface—the locus of points where the surface is tangent to the lines of sight originating from the camera's center of projection. When the surface moves, the contour generator moves as well and it's motion is constrained both by the rigid motion of the surface and by the relative position of the surface with respect to the camera. Therefore, the contour generator has two motion components and we must explicitly estimate these two components.

First, we will determine the constraints that formally define the contour generator. The extremal contour is simply determined by projecting the contour generator onto the image plane. Second, we will derive a closed-form solution for the *extremal-contour Jacobian*, i.e., the Jacobian matrix that maps 3-D joint velocities onto 2-D contour-point velocities.

2.2 The Contour-Generator Constraint and Extremal Contours

Let X be a 3-D point that lies onto the smooth surface of a body part, and let $\boldsymbol{X} = (X_1, X_2, X_3)$ be the coordinates of this point in the body-part frame, Fig. 2. Without loss of generality, the camera frame will be chosen to be identical to the world frame. Hence, the world coordinates of X are:

$$\boldsymbol{X}^w = \mathbf{R}(\boldsymbol{\Phi})\boldsymbol{X} + \boldsymbol{t}(\boldsymbol{\Phi}). \tag{2}$$

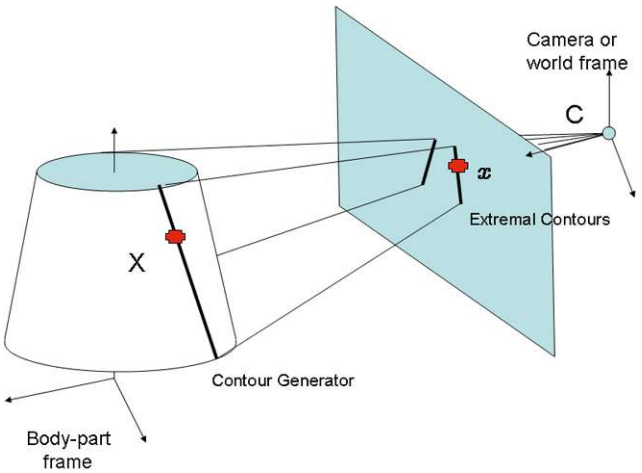


Fig. 2 A truncated elliptical cone is an example of a developable surface used to model a body part. Such a surface projects onto an image as a pair of *extremal contours*. The 2-D motion of these extremal contours is a function of both the motion of the body-part itself as well as the sliding of the *contour generator* along the smooth surface of the part

The contour generator is the locus of points lying onto the surface where the lines of sight (originating at the optical center of the camera and passing through image points) are tangent to that surface. Obviously, the contour generator is defined by:

$$(\mathbf{R}\boldsymbol{n})^\top (\mathbf{R}\boldsymbol{X} + \boldsymbol{t} - \boldsymbol{C}) = 0 \tag{3}$$

where the surface normal $\boldsymbol{n}$ is defined by the following cross-product:

$$\boldsymbol{n} = \frac{\partial \boldsymbol{X}}{\partial z} \times \frac{\partial \boldsymbol{X}}{\partial \theta} = \boldsymbol{X}_z \times \boldsymbol{X}_\theta. \tag{4}$$

Here the couple (z, θ) is a parameterization of the body-part's surface and $\boldsymbol{C}$ denotes the camera's optical center. The equation above becomes:

$$\boldsymbol{X}^\top \boldsymbol{n} + (\boldsymbol{t} - \boldsymbol{C})^\top \mathbf{R}\boldsymbol{n} = 0 \tag{5}$$

or:

$$(\boldsymbol{X} + \boldsymbol{m})^\top \boldsymbol{n} = 0 \tag{6}$$

with $\boldsymbol{m} = \mathbf{R}^\top(\boldsymbol{t} - \boldsymbol{C})$. Equation (6) is the contour-generator constraint that must be satisfied at each time instant. Once the contour generator is determined, the 2-D extremal contour (the projection of the contour generator) can be found in the camera frame from:

$$\begin{pmatrix} s\boldsymbol{x} \\ s \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{X}^w \\ 1 \end{pmatrix}. \tag{7}$$

2.3 The Contour Generator of a Developable Surface

It would be difficult to treat the general case of curved surfaces. An interesting case is the class of developable surfaces which are a special case of ruled surfaces (Do Carmo 1976). We prove the following result:

Proposition *Under perspective projection, the contour generators of a developable surface are rulings of the surface, i.e., they are line segments.*

This also means that the extremal contours of a developable surface are straight lines. In practice we need to consider surfaces that are well suited to model body parts. We will use elliptical cones but the result of this section allows one to use any kind of developable surfaces.

Consider a differentiable one-parameter family of straight lines $(\boldsymbol{\alpha}(\theta), \boldsymbol{\beta}(\theta))$ where to each θ are assigned a 3-D point $\boldsymbol{\alpha}(\theta)$ and a 3-D vector $\boldsymbol{\beta}(\theta)$, so that both $\boldsymbol{\alpha}(\theta)$ and $\boldsymbol{\beta}(\theta)$ depend differentiably on θ. The parametrized surface:

$$\boldsymbol{X}(\theta, z) = \boldsymbol{\alpha}(\theta) + z\boldsymbol{\beta}(\theta) \tag{8}$$

is called a ruled surface, and the normal to this surface is given by (4):

$$\boldsymbol{n} = \boldsymbol{X}_\theta \times \boldsymbol{X}_z = (\boldsymbol{\alpha}' + z\boldsymbol{\beta}') \times \boldsymbol{\beta}. \tag{9}$$

Since a developable surface is a ruled surface whose Gaussian curvature is null everywhere on the surface, one can show ((Do Carmo 1976), (Kreyzig 1991)) that the normal to this surface can be written as:

$$\boldsymbol{n} = (1 + bz)\boldsymbol{\beta}' \times \boldsymbol{\beta}. \tag{10}$$

Notice that the direction of the normal is given by the cross-product of $\boldsymbol{\beta}'$ and $\boldsymbol{\beta}$ and it depends only on the parameter θ. Using this parameterization of the normal, we can rewrite the contour generator constraint, (6), for developable surfaces as follows:

$$(\boldsymbol{\alpha}(\theta) + \boldsymbol{m})^\top (\boldsymbol{\beta}'(\theta) \times \boldsymbol{\beta}(\theta)) = 0. \tag{11}$$

One should notice that this contour-generator constraint involves only the surface parameter θ and not the z parameter. Therefore, any solution of (11), say $\hat{\theta}$, will yield the entire ruling line $\boldsymbol{X}(\hat{\theta}, z)$. This proves that under perspective projection, the contour generators of a developable surface are rulings of the surface, i.e. line segments. As a result, *the kinematics of the contour generators are fully determined by the evolution of the solutions $\hat{\theta}(t)$ of the contour generator equation over time*.

2.4 Truncated Elliptical Cones

In practice will model body parts with truncated elliptical cones. Such a shape is bounded by two planar faces which produce discontinuity contours, as well as a curved surface which produces a pair of extremal contours. The latter can be easily parameterized in cylindrical coordinates by an angle θ and a height z as a ruled surface:

$$\boldsymbol{X}(\theta, z) = \begin{pmatrix} a\cos\theta \\ b\sin\theta \\ 0 \end{pmatrix} + z \begin{pmatrix} ak\cos\theta \\ bk\sin\theta \\ 1 \end{pmatrix} \tag{12}$$

where a and b are minor and major half-axes of the elliptical cross-section, k is the tapering parameter of the cone and $z \in [z_1, z_2]$. It is straightforward to verify that an elliptical cone is a developable surface. Below we provide an analytical expression of its associated contour generators.

With this parametrization, (11) can be easily expanded to yield a trigonometric constraint of the form $F\cos\theta + G\sin\theta + H = 0$ where F, G and H depend on $\mathbf{R}(\boldsymbol{\Phi})$, $\boldsymbol{t}(\boldsymbol{\Phi})$ and C while they are independent of the parameter z. In order to solve this equation and find its roots we use the standard trigonometric substitution, i.e., $\tan\frac{\theta}{2}$ and obtain a second-degree polynomial:

$$(H - F)\tan^2\frac{\theta}{2} + 2G\tan\frac{\theta}{2} + (F + H) = 0. \tag{13}$$

This equation has two real solutions, θ_1 and θ_2, whenever the camera's optical center lies outside the cone that defines the body part (a constrained that is rarely violated). Therefore, in the case of elliptical cones the contour generator is composed of two straight lines parameterized by z, i.e., $X(\boldsymbol{\Phi}, \theta_1, z)$ and $X(\boldsymbol{\Phi}, \theta_2, z)$.

2.5 The Motion of Extremal Contours

We turn our attention back to extremal contours—the projection onto the image plane of the contour generator. We denote by $\boldsymbol{x} = (x_1, x_2)$ the real-valued image coordinates of an extremal-contour point, i.e., (7). The motion of this point depends on both:

- The *rigid motion* of the body-part with respect to the world reference frame, and
- the *sliding motion* of the contour generator onto the part's curved surface, as the relative position and orientation of this part varies with respect to the camera.

We formally derive the motion of an extremal contour point in terms of these two components. The 2-D velocity of an extremal-contour point is:

$$\frac{d\boldsymbol{x}}{dt} = \frac{d\boldsymbol{x}}{d\boldsymbol{X}^w}\frac{d\boldsymbol{X}^w}{d\boldsymbol{r}}\frac{d\boldsymbol{r}}{d\boldsymbol{\Phi}}\frac{d\boldsymbol{\Phi}}{dt}. \tag{14}$$

Vector $\boldsymbol{X}^w$, already defined by (2), denotes the contour-generator point in world coordinates. Its projection is obtained from (7):

$$x_1 = \frac{X_1^w}{X_3^w}, \qquad x_2 = \frac{X_2^w}{X_3^w}. \tag{15}$$

We recall that $\boldsymbol{r}$ was already defined in Sect. 2.1 and it denotes the pose parameters associated with the body-part. Since the latter is linked to the root part by a kinematic chain, $\boldsymbol{r}$ is in it's turn parameterized by $\boldsymbol{\Phi}$. We have:

- The first term of the right-hand side of (14) is the image Jacobian denoted by $\mathbf{J}_I$:

$$\frac{d\boldsymbol{x}}{d\boldsymbol{X}^w} = \mathbf{J}_I = \begin{bmatrix} 1/X_3^w & 0 & -X_1^w/(X_3^w)^2 \\ 0 & 1/X_3^w & -X_2^w/(X_3^w)^2 \end{bmatrix}. \tag{16}$$

- The second term is a transformation that allows to determine the velocity of a point from the motion of the part on which this point lies. When the point is rigidly attached to the part, this transformation is given by matrix $\mathbf{A}$ (see below). When the point slides onto the smooth surface there

is a second transformation— matrix $\mathbf{B}$—that remains to be determined:

$$\frac{d\boldsymbol{X}^w}{d\boldsymbol{r}} = \mathbf{A} + \mathbf{B}. \tag{17}$$

- The third term is the Jacobian of the kinematic chain that links the body part to a root body part and to a world reference frame. This Jacobian matrix will be denoted by $\mathbf{J}_H$.
- The fourth term is the vector composed of both the joint velocities and the velocity of the root body part.

With these notations, (14) becomes:

$$\dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{d\boldsymbol{\Phi}}\dot{\boldsymbol{\Phi}} \tag{18}$$

where:

$$\frac{d\boldsymbol{x}}{d\boldsymbol{\Phi}} = \mathbf{J}_I(\mathbf{A} + \mathbf{B})\mathbf{J}_H \tag{19}$$

is the extremal-contour Jacobian that will be used by the tracker. It is useful to introduce the kinematic-screw notation, i.e., a six dimensional vector concatenating the rotational velocity, $\boldsymbol{\Omega}$, and the translational velocity, $\boldsymbol{V}$ (see below):

$$\frac{d\boldsymbol{r}}{dt} = \begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix}. \tag{20}$$

The velocity of an extremal-contour point can therefore be written as:

$$\dot{\boldsymbol{x}} = \mathbf{J}_I(\mathbf{A} + \mathbf{B}) \begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix}. \tag{21}$$

Let us now make explicit the 3×6 matrices $\mathbf{A}$ and $\mathbf{B}$. By differentiation of (2), we obtain:

$$\dot{\boldsymbol{X}}^w = \dot{\mathbf{R}}\boldsymbol{X} + \dot{\boldsymbol{t}} + \mathbf{R}\dot{\boldsymbol{X}}. \tag{22}$$

Equation (22) reveals that unlike the motion of a point that is rigidly attached to a surface, the motion of a contour-generator point has two components:

- A component due to the rigid motion of the smooth surface, $\dot{\mathbf{R}}\boldsymbol{X} + \dot{\boldsymbol{t}}$, and
- a component due to the sliding of the contour generator onto this smooth surface, $\mathbf{R}\dot{\boldsymbol{X}}$.

2.5.1 The Rigid-Motion Component

The first component in (22) can be parameterized by the kinematic screw and it becomes:

$$\dot{\mathbf{R}}\boldsymbol{X} + \dot{\boldsymbol{t}} = \dot{\mathbf{R}}\mathbf{R}^\top(\boldsymbol{X}^w - \boldsymbol{t}) + \dot{\boldsymbol{t}} = \mathbf{A}\begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix} \tag{23}$$

with $[\boldsymbol{\Omega}]_\times = \dot{\mathbf{R}}\mathbf{R}^\top$, $\dot{\boldsymbol{t}} = \boldsymbol{V}$, and where $\mathbf{A}$ is the 3×6 matrix that allows to compute the velocity of a point from the kinematic screw of the rigid-body motion:

$$\mathbf{A} = [[\boldsymbol{t} - \boldsymbol{X}^w]_\times \ \mathbf{I}_{3\times 3}]. \tag{24}$$

The notation $[\boldsymbol{m}]_\times$ stands for the 3×3 skew-symmetric matrix associated with the 3-vector $\boldsymbol{m}$. Vectors $\boldsymbol{\Omega}$ and $\boldsymbol{V}$ can be concatenated to form a 6-vector $(\boldsymbol{\Omega}\ \boldsymbol{V})^\top$ which is known as the kinematic screw—the rotational and translational velocities of the body part in world coordinates. This factorization is strictly equivalent with $\boldsymbol{V} = \dot{\boldsymbol{t}} - \dot{\mathbf{R}}\mathbf{R}^\top\boldsymbol{t}$ and $\mathbf{A} = [[\boldsymbol{X}^w]_\times \ \mathbf{I}]$.

2.5.2 The Sliding-Motion Component

It is interesting to notice that, although the link between image contours and smooth surfaces has been thoroughly studied in the past, the problem of inferring the velocity of these contours when the smooth surface undergoes a general 3-D motion has not yet been addressed. In the general case, the sliding-motion component is a complex function of both the local surface shape and of the relative motion between the surface and the observer. The problem is strongly linked to the problem of computing the aspects of a smooth surface (Koenderink 1990) and (Forsyth and Ponce 2003) (Chaps. 19 and 20). Both these textbooks treat the case of a static object viewed under orthographic projection.

We establish a mathematical formalism for developable surfaces, i.e., (Do Carmo 1976) when they are viewed under perspective projection. As it has been shown above, the contour generators are rulings of the surface and their motion are fully determined by computing the time derivatives of their θ parameters. If a surface point $\boldsymbol{X}$ lies onto the contour generator, then its observed sliding velocity is:

$$\dot{\boldsymbol{X}} = \frac{\partial \boldsymbol{X}}{\partial \theta}\dot{\theta} + \frac{\partial \boldsymbol{X}}{\partial z}\dot{z} = \boldsymbol{X}_\theta\,\dot{\theta} + \boldsymbol{X}_z\,\dot{z}. \tag{25}$$

The sliding velocity along the contour generator itself, $\dot{z}$, is not observable because the contour generator is the ruling of the surface—a straight line. Therefore one may assume that:

$$\dot{z} = 0. \tag{26}$$

Therefore, the sliding-motion component in (22) can be written as:

$$\mathbf{R}\dot{\boldsymbol{X}} = \mathbf{R}\boldsymbol{X}_\theta\,\dot{\theta}. \tag{27}$$

Since $\boldsymbol{X}$ lies onto the contour generator, it verifies the contour generator constraint, i.e., (5). By differentiation of this equation we obtain a constraint for the surface parameter velocity, $\dot{\theta}$, as follows. We differentiate equation (5), we perform the substitutions $\dot{\mathbf{R}}^\top = -\mathbf{R}^\top[\boldsymbol{\Omega}]_\times$ and $\dot{\boldsymbol{t}} = \boldsymbol{V}$, and we

notice that the velocity of a surface point is tangent to the surface, i.e., $\dot{\boldsymbol{X}}^{\top}\boldsymbol{n}=0$. We obtain the following expression for the derivative of (5):

$$(\boldsymbol{X}+\mathbf{R}^{\top}(\boldsymbol{t}-\boldsymbol{C}))^{\top}\dot{\boldsymbol{n}}=([\boldsymbol{\Omega}]_{\times}(\boldsymbol{t}-\boldsymbol{C})-\boldsymbol{V})^{\top}\mathbf{R}\boldsymbol{n}. \tag{28}$$

With $\dot{\boldsymbol{n}}=\boldsymbol{n}_{\theta}\dot{\theta}$ and with $[\boldsymbol{a}]_{\times}\boldsymbol{b}=-[\boldsymbol{b}]_{\times}\boldsymbol{a}$, we obtain from (28):

$$\dot{\theta}=\frac{(\mathbf{R}\boldsymbol{n})^{\top}[[\boldsymbol{C}-\boldsymbol{t}]_{\times}-\mathbf{I}_{3\times3}]}{(\boldsymbol{X}+\mathbf{R}^{\top}(\boldsymbol{t}-\boldsymbol{C}))^{\top}\boldsymbol{n}_{\theta}}\begin{pmatrix}\boldsymbol{\Omega}\\ \boldsymbol{V}\end{pmatrix}. \tag{29}$$

Therefore, the sliding velocity $\dot{\theta}$ can be expressed as a function of (i) the surface parameterization, (ii) the relative position and orientation of the camera with respect to the surface, and (iii) the rigid motion of the surface (the kinematic screw). To summarize, (27) becomes:

$$\mathbf{R}\dot{\boldsymbol{X}}=\mathbf{B}\begin{pmatrix}\boldsymbol{\Omega}\\ \boldsymbol{V}\end{pmatrix} \tag{30}$$

where $\mathbf{B}$ is the 3×6 matrix:

$$\mathbf{B}=\frac{1}{b}\mathbf{R}\boldsymbol{X}_{\theta}(\mathbf{R}\boldsymbol{n})^{\top}[[\boldsymbol{C}-\boldsymbol{t}]_{\times}\ \ -\mathbf{I}_{3\times3}] \tag{31}$$

and the scalar b is defined by:

$$b=(\boldsymbol{X}+\mathbf{R}^{\top}(\boldsymbol{t}-\boldsymbol{C}))^{\top}\boldsymbol{n}_{\theta}.$$

2.5.3 The Velocity of Extremal Contours

To conclude this section, the velocity of an extremal contour point has a rigid-motion component and a surface-sliding component:

$$\dot{\boldsymbol{x}}=\dot{\boldsymbol{x}}^{r}+\dot{\boldsymbol{x}}^{s}. \tag{32}$$

The explicit parameterization of the sliding component, as shown above, allows its incorporation into the explicit representation of the observed image velocities as a function of the kinematic-chain parameters, as described in detail below.

The sliding velocity depends both on the curvature of the surface and on the velocity of the surface. In practice it will speed up the convergence of the tracker by a factor of two, as described in Sect. 6.

3 The Human-Body Kinematic Chain

In the case of a kinematic chain, the rigid motion of a body-part can be parameterized by the joint parameters. Kinematic chains are widely used by human-body trackers and motion capture systems. In this section we introduce the use

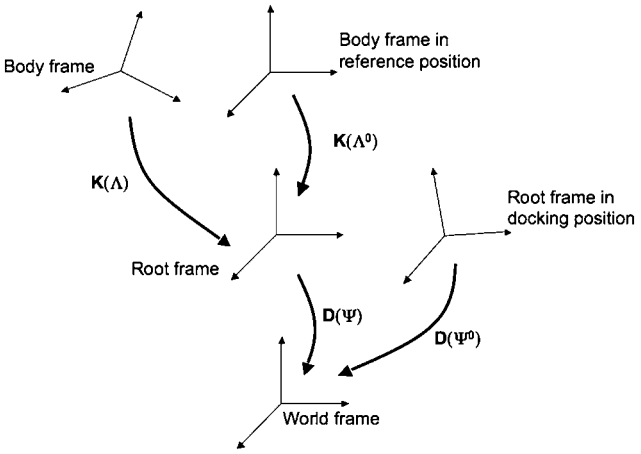


Fig. 3 Each body part has a frame associated with it, therefore motions are represented by changes in coordinate frames. There is a reference position for each body part in the chain defined by the joint angles $\boldsymbol{\Lambda}^{0}$. Similarly, there is a docking position for the root body-part defined by the six-dimensional vector $\boldsymbol{\Psi}^{0}$

of the *zero-reference kinematic representation* for modeling the human-body articulated chains. The Zero-reference kinematic representation was studied for robot manipulators (Mooring et al. 1991) and (McCarthy 1990). The parameterization introduced in this section combines the zero-reference representation with the free motion of the root body-part, i.e., Fig. 3.

Without loss of generality we consider any one among the several kinematic chains needed to describe the human body. A body part P is linked to the root body-part R by a kinematic chain with p rotational degrees of freedom. The root body part itself moves freely with six degrees of freedom (three rotations and three translations) and with respect to the world coordinate frame. Let $\boldsymbol{\Lambda}=(\lambda_1,\ldots,\lambda_p)$ denote the joint angles associated with the kinematic chain, and let $\boldsymbol{\Psi}=(\psi_1,\ldots,\psi_q)$ denote the rotational and translational degrees of freedom of the free motion. In the most general case we have $q=6$. Therefore, there are $p+q$ motion parameters embedded in the vector $\boldsymbol{\Phi}=(\boldsymbol{\Psi},\boldsymbol{\Lambda})$.

With the same notations as in the previous section, we consider a point X that belongs to the contour generator associated with a developable surface and body part. The point's *homogeneous coordinates* in the local frame are denoted by $\widetilde{\boldsymbol{X}}=(X_1\ X_2\ X_3\ 1)^{\top}$. We also denote by $\boldsymbol{X}^{r}$ the coordinates of the same point in the root body-part frame, and by $\boldsymbol{X}^{w}$ its coordinates in the world frame.

Moreover, we denote with $\mathbf{D}(\boldsymbol{\Psi})$ the 4×4 homogeneous matrix associated with the free motion of the root body part with respect to a fixed world frame, and with $\mathbf{K}(\boldsymbol{\Lambda})$ the 4×4 homogeneous matrix associated with the constrained motion of a body part with respect to the root part. Let $\boldsymbol{\Lambda}^{0}$ be the joint angles for a particular *reference position* of the kinematic chain. Obviously we have

$\widetilde{\boldsymbol{X}}^r(\boldsymbol{\Lambda}) = \mathbf{K}(\boldsymbol{\Lambda})\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{X}}^r(\boldsymbol{\Lambda}^0) = \mathbf{K}(\boldsymbol{\Lambda}^0)\widetilde{\boldsymbol{X}}$. We obtain $\widetilde{\boldsymbol{X}}^r(\boldsymbol{\Lambda}) = \mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda}^0)\widetilde{\boldsymbol{X}}^r(\boldsymbol{\Lambda}^0)$. With this formula and from $\widetilde{\boldsymbol{X}}^w(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \mathbf{D}(\boldsymbol{\Psi})\widetilde{\boldsymbol{X}}^r(\boldsymbol{\Lambda})$ we obtain:

$$\widetilde{\boldsymbol{X}}^w(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \mathbf{D}(\boldsymbol{\Psi})\mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda}^0)\widetilde{\boldsymbol{X}}^r(\boldsymbol{\Lambda}^0).$$

We also consider a reference or a *docking* position for the root body-part, defined by the free-motion parameters $\boldsymbol{\Psi}^0$, i.e., $\widetilde{\boldsymbol{X}}^w(\boldsymbol{\Psi}^0, \boldsymbol{\Lambda}^0) = \mathbf{D}(\boldsymbol{\Psi}^0)\widetilde{\boldsymbol{X}}^r(\boldsymbol{\Lambda}^0)$. Finally we obtain:

$$\begin{aligned}&\widetilde{\boldsymbol{X}}^w(\boldsymbol{\Psi}, \boldsymbol{\Lambda})\\ &\quad= \mathbf{D}(\boldsymbol{\Psi})\mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda}^0)\mathbf{D}^{-1}(\boldsymbol{\Psi}^0)\widetilde{\boldsymbol{X}}^w(\boldsymbol{\Psi}^0, \boldsymbol{\Lambda}^0) \quad (33)\\ &\quad= \mathbf{H}(\boldsymbol{\Psi}, \boldsymbol{\Psi}^0, \boldsymbol{\Lambda}, \boldsymbol{\Lambda}^0)\widetilde{\boldsymbol{X}}^w(\boldsymbol{\Psi}^0, \boldsymbol{\Lambda}^0). \quad (34)\end{aligned}$$

It will be convenient to write the above transformation as:

$$\mathbf{H}(\boldsymbol{\Psi}, \boldsymbol{\Psi}^0, \boldsymbol{\Lambda}, \boldsymbol{\Lambda}^0) = \mathbf{F}(\boldsymbol{\Psi}, \boldsymbol{\Psi}^0)\,\mathbf{Q}(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^0, \boldsymbol{\Psi}^0) \quad (35)$$

with:

$$\mathbf{F}(\boldsymbol{\Psi}, \boldsymbol{\Psi}^0) = \mathbf{D}(\boldsymbol{\Psi})\mathbf{D}^{-1}(\boldsymbol{\Psi}^0) \quad (36)$$

and:

$$\mathbf{Q}(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^0, \boldsymbol{\Psi}^0) = \mathbf{D}(\boldsymbol{\Psi}^0)\mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda}^0)\mathbf{D}^{-1}(\boldsymbol{\Psi}^0). \quad (37)$$

3.1 The Kinematic-Chain Model

The transformation $\mathbf{K}$ describes an open kinematic chain and the transformation $\mathbf{Q}$ describes exactly the same chain but relatively to a *reference position* of the chain. $\mathbf{K}$ may be written as a composition of fixed transformations $\mathbf{L}_1, \ldots, \mathbf{L}_p$, and of one-degree-of-freedom rotations $\mathbf{J}(\lambda_1), \ldots, \mathbf{J}(\lambda_p)$:

$$\mathbf{K}(\boldsymbol{\Lambda}) = \mathbf{L}_1\mathbf{J}(\lambda_1)\cdots\mathbf{L}_p\mathbf{J}(\lambda_p) \quad (38)$$

where the matrices $\mathbf{L}_1 \ldots \mathbf{L}_p$ are fixed transformations between adjacent rotational joints, and matrices of the form of $\mathbf{J}$ are the canonical representations of a rotation.

Matrix $\mathbf{Q}$ in (37) can now be written as a product of one-degree-of-freedom transformations $\mathbf{Q}_i$:

$$\begin{aligned}\mathbf{Q}(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^0, \boldsymbol{\Psi}^0) = \mathbf{Q}_1(\lambda_1 - \lambda_1^0)\cdots\mathbf{Q}_i(\lambda_i - \lambda_i^0)\cdots\\ \times\, \mathbf{Q}_p(\lambda_p - \lambda_p^0) \quad (39)\end{aligned}$$

where each term $\mathbf{Q}_i$ is of the form $\mathbf{U}_i\mathbf{J}(\lambda_i - \lambda_i^0)\mathbf{U}_i^{-1}$, i.e., (McCarthy 1990):

$$\begin{aligned}\mathbf{Q}_i(\lambda_i - \lambda_i^0) = \underbrace{\mathbf{D}(\boldsymbol{\Psi}^0)\mathbf{L}_1\mathbf{J}(\lambda_1^0)\cdots\mathbf{L}_i}_{\mathbf{U}_i}\,\mathbf{J}(\lambda_i - \lambda_i^0)\\ \times \underbrace{\mathbf{L}_i^{-1}\cdots\mathbf{J}(-\lambda_1^0)\mathbf{L}_1^{-1}\mathbf{D}^{-1}(\boldsymbol{\Psi}^0)}_{\mathbf{U}_i^{-1}}. \quad (40)\end{aligned}$$

Notice that matrices $\mathbf{U}_i, \{i = 1 \ldots p\}$ remain fixed when the joint parameters vary **and** when the root body-part undergoes a free motion. Others used the exponential representation for this one-dimensional transformations (Murray et al. 1994; Bregler et al. 2004).

3.2 The Zero-Reference Kinematic Model

Without loss of generality, one may set the initial joint-angle values to zero, i.e., $\lambda_1^0 = \cdots = \lambda_p^0 = 0$. In this case, the kinematic chain does not depend any more on its reference pose, since $\mathbf{J}(\lambda_i^0) = \mathbf{I}$ for all i. The kinematic chain writes in this case:

$$\mathbf{Q}(\boldsymbol{\Lambda}, \boldsymbol{\Psi}^0) = \mathbf{Q}_1(\lambda_1, \boldsymbol{\Psi}^0)\cdots\mathbf{Q}_i(\lambda_i, \boldsymbol{\Psi}^0)\cdots\mathbf{Q}_p(\lambda_p, \boldsymbol{\Psi}^0). \quad (41)$$

The human-body zero-reference kinematic chain From the equations above, one may write a compact and convenient factorization of matrix $\mathbf{H}$, i.e., (35):

$$\begin{aligned}\mathbf{H}(\boldsymbol{\Psi}, \boldsymbol{\Psi}^0, \boldsymbol{\Lambda}) = \mathbf{F}(\boldsymbol{\Psi}, \boldsymbol{\Psi}^0)\mathbf{Q}_1(\lambda_1, \boldsymbol{\Psi}^0)\cdots\\ \times\, \mathbf{Q}_i(\lambda_i, \boldsymbol{\Psi}^0)\cdots\mathbf{Q}_p(\lambda_p, \boldsymbol{\Psi}^0). \quad (42)\end{aligned}$$

4 The Jacobian of the Human-Body Kinematic Chain

In this section we make explicit the Jacobian matrix associated with the kinematic chain of the human-body, $\mathbf{J}_H$. This matrix appears in (14); From this equation and from (21) we obtain:

$$\begin{pmatrix}\boldsymbol{\Omega}\\ \boldsymbol{V}\end{pmatrix} = \mathbf{J}_H\dot{\boldsymbol{\Phi}}. \quad (43)$$

The Jacobian of a kinematic chain such as the one described above is intrinsic to the mechanical and geometric structure of the kinematic chain and it does not depend on a particular choice of a point X, is it sliding onto the surface or rigidly attached to it. The Jacobian $\mathbf{J}_H$ maps joint velocities onto the kinematic screw of a body part whose kinematic chain is denoted by $\mathbf{H}$. In order to establish an expression for the Jacobian, we will first need to determine the tangent operator $\widehat{\mathbf{H}}$ of $\mathbf{H}$:

$$\widehat{\mathbf{H}} = \dot{\mathbf{H}}\mathbf{H}^{-1}. \quad (44)$$

Second, we parameterize $\widehat{\mathbf{H}}$ such that it depends only on the kinematic parameters, i.e., we must take the derivative of a body-part point with respect to the motion variables, i.e., $d\boldsymbol{X}^w/d\boldsymbol{\Phi}$. The case of human-body motion is different than the classical case studied in the robotics literature, (McCarthy 1990; Mooring et al. 1991; Murray et al. 1994) because one must take into account the fact that the root-part of the chain undergoes a free rigid motion.

4.1 A Rotational Joint

First, we consider the case of a single rotational joint. It's tangent operator is defined by $\widehat{\mathbf{Q}}_i = \dot{\mathbf{Q}}_i \mathbf{Q}_i^{-1}$, and we obviously have $\widehat{\mathbf{Q}}_i = \mathbf{U}_i \widehat{\mathbf{J}}_i \mathbf{U}_i^{-1}$. From $\widehat{\mathbf{J}} = \dot{\mathbf{J}}\mathbf{J}^{-1}$, we have:

$$\widehat{\mathbf{J}}(\lambda_i) = \dot{\lambda}_i \widetilde{\mathbf{J}},$$

with

$$\widetilde{\mathbf{J}} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Therefore, we obtain a simple expression for the tangent operator associated with one joint:

$$\widehat{\mathbf{Q}}_i(\lambda_i - \lambda_i^0) = \dot{\lambda}_i \mathbf{U}_i \widetilde{\mathbf{J}} \mathbf{U}_i^{-1} = \dot{\lambda}_i \widetilde{\mathbf{Q}}_i. \tag{45}$$

Matrix $\widetilde{\mathbf{J}}$ is called the *Lie-algebra of the Lie-group* defined by the matrices of the form of $\mathbf{J}$. If one prefers the exponential representation, $\widetilde{\mathbf{J}}$ is called a *twist*.

4.2 The Tangent Operator

Second, we determine the tangent operator of the human-body kinematic chain. The zero-reference kinematic chain, $\mathbf{H}$, may well be viewed as an Euclidean transformation and is composed of a rotation matrix and a translation vector: $\mathbf{R}_H, \boldsymbol{t}_H$. Hence, its tangent operator has a rigid-motion component, i.e., (23) and (24), as well as a sliding-motion component, i.e., (30). When applied to (34) we obtain *the action of the tangent operator* onto a surface point:

$$\begin{aligned} \dot{\boldsymbol{X}}^w(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) &= \dot{\mathbf{R}}_H \boldsymbol{X}^w(\boldsymbol{\Psi}_0, \boldsymbol{\Lambda}_0) + \dot{\boldsymbol{t}}_H + \mathbf{R}_H \dot{\boldsymbol{X}}^w(\boldsymbol{\Psi}_0, \boldsymbol{\Lambda}_0) \\ &= \dot{\mathbf{R}}_H \mathbf{R}_H^\top (\boldsymbol{X}^w(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) - \boldsymbol{t}_H) + \dot{\boldsymbol{t}}_H \\ &\quad + \mathbf{R}_H \dot{\boldsymbol{X}}^w(\boldsymbol{\Psi}_0, \boldsymbol{\Lambda}_0) \\ &= (\mathbf{A}_H + \mathbf{B}_H) \begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix} \end{aligned} \tag{46}$$

where we have $\boldsymbol{\Omega} = \dot{\mathbf{R}}_H \mathbf{R}_H^\top$, $\boldsymbol{V} = \dot{\boldsymbol{t}}_H$ and with 3×6 matrices $\mathbf{A}_H$ and $\mathbf{B}_H$ as defined by (24) and (31). The 3-D vectors $\boldsymbol{\Omega}$ and $\boldsymbol{V}$ form the kinematic screw which we seek to estimate:

$$\widehat{\mathbf{H}}(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \begin{bmatrix} [\boldsymbol{\Omega}]_\times & \boldsymbol{V} \\ \mathbf{0}^\top & 0 \end{bmatrix}. \tag{47}$$

Since $\mathbf{H} = \mathbf{F}\mathbf{Q}$, we have $\dot{\mathbf{H}} = \dot{\mathbf{F}}\mathbf{Q} + \mathbf{F}\dot{\mathbf{Q}}$ and:

$$\widehat{\mathbf{H}}(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \widehat{\mathbf{F}}(\boldsymbol{\Psi}) + \mathbf{F}\widehat{\mathbf{Q}}(\boldsymbol{\Lambda})\mathbf{F}^{-1}. \tag{48}$$

As detailed below, the tangent operator can be written as the sum:

$$\widehat{\mathbf{H}}(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \widehat{\mathbf{H}}_r(\boldsymbol{\Psi}) + \sum_{i=1}^{p} \widehat{\mathbf{H}}_i(\lambda_i). \tag{49}$$

- *The tangent operator associated with the free motion of the root body-part*, $\widehat{\mathbf{H}}_r(\boldsymbol{\Psi}) = \widehat{\mathbf{F}}(\boldsymbol{\Psi})$; from (36) we obtain: $\widehat{\mathbf{F}}(\boldsymbol{\Psi}) = \widehat{\mathbf{D}}(\boldsymbol{\Psi})$. $\widehat{\mathbf{H}}_r$ is the 4×4 matrix parameterized by the rotational velocity $\boldsymbol{\omega}_r$ and the translational velocity $\boldsymbol{v}_r$ of this free motion:

$$\widehat{\mathbf{H}}_r(\boldsymbol{\Psi}) = \begin{bmatrix} [\boldsymbol{\omega}_r]_\times & \boldsymbol{v}_r \\ \mathbf{0}^\top & 0 \end{bmatrix}. \tag{50}$$

  This motion has six degrees of freedom and can be parameterized by three rotations and three translations:

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_z(\psi_3)\mathbf{R}_y(\psi_2)\mathbf{R}_x(\psi_1), \\ \boldsymbol{t} &= \psi_4 \boldsymbol{e}_x + \psi_5 \boldsymbol{e}_y + \psi_6 \boldsymbol{e}_z, \end{aligned}$$

  where $\boldsymbol{e}_x = (1\,0\,0)^\top$ and so forth. The kinematic screw of this motion can therefore be written as:

$$\begin{pmatrix} \boldsymbol{\omega}_r \\ \boldsymbol{v}_r \end{pmatrix} = \begin{bmatrix} \boldsymbol{\omega}_x & \boldsymbol{\omega}_y & \boldsymbol{\omega}_z & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z \end{bmatrix}_{6\times 6} \dot{\boldsymbol{\Psi}} \tag{51}$$

  with $[\boldsymbol{\omega}_z]_\times = [\boldsymbol{e}_z]_\times$, $[\boldsymbol{\omega}_y]_\times = \mathbf{R}_z[\boldsymbol{e}_y]_\times \mathbf{R}_z^\top$, and $[\boldsymbol{\omega}_x]_\times = \mathbf{R}_y\mathbf{R}_z[\boldsymbol{e}_x]_\times \mathbf{R}_z^\top \mathbf{R}_y^\top$.
- *The tangent operator associated with the constrained motion of the kinematic chain*, $\mathbf{F}\widehat{\mathbf{Q}}(\boldsymbol{\Lambda})\mathbf{F}^{-1}$; it is expressed in world coordinates and with respect to a reference position defined by both $\boldsymbol{\Psi}^0$ and $\boldsymbol{\Lambda}^0$. This tangent operator can be expanded as (McCarthy 1990):

$$\mathbf{F}\widehat{\mathbf{Q}}(\boldsymbol{\Lambda})\mathbf{F}^{-1} = \begin{bmatrix} \boldsymbol{\omega}_1 & \dots & \boldsymbol{\omega}_p \\ \boldsymbol{v}_1 & \dots & \boldsymbol{v}_p \end{bmatrix} \dot{\boldsymbol{\Lambda}}, \tag{52}$$

  with $\dot{\boldsymbol{\Lambda}} = (\dot{\lambda}_1 \dots \dot{\lambda}_p)^\top$.

Therefore, by combining (49), (51), and (52) we obtain the following expression for the kinematic screw:

$$\begin{aligned} \begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{pmatrix} &= \begin{bmatrix} \boldsymbol{\omega}_x & \boldsymbol{\omega}_y & \boldsymbol{\omega}_z & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z \end{bmatrix} \dot{\boldsymbol{\Psi}} \\ &\quad + \begin{bmatrix} \boldsymbol{\omega}_1 & \dots & \boldsymbol{\omega}_p \\ \boldsymbol{v}_1 & \dots & \boldsymbol{v}_p \end{bmatrix} \dot{\boldsymbol{\Lambda}} \end{aligned} \tag{53}$$

with $\dot{\boldsymbol{\Psi}} = (\dot{\psi}_1 \dots \dot{\psi}_6)^\top$ and $\dot{\boldsymbol{\Lambda}} = (\dot{\lambda}_1 \dots \dot{\lambda}_p)^\top$. Finally, the Jacobian of the human-body kinematic chain writes as

a $6 \times (6+p)$ matrix:

$$\mathbf{J}_H = \begin{bmatrix} \boldsymbol{\omega}_x & \boldsymbol{\omega}_y & \boldsymbol{\omega}_z & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\omega}_1 & \dots & \boldsymbol{\omega}_p \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z & \boldsymbol{v}_1 & \dots & \boldsymbol{v}_p \end{bmatrix}. \quad (54)$$

To conclude this section we remind that the relationship between the kinematic velocities $\dot{\boldsymbol{\Phi}} = (\dot{\boldsymbol{\Psi}}\ \dot{\boldsymbol{\Lambda}})$ and the image velocity of an extremal contour point x writes:

$$\dot{\boldsymbol{x}} = \mathbf{J}_I(\mathbf{A}_H + \mathbf{B}_H)\mathbf{J}_H\dot{\boldsymbol{\Phi}}. \quad (55)$$

This corresponds to (21) where the kinematic screw is given by (53).

5 Fitting Extremal Contours to Image Contours

In this section we consider the problem of fitting extremal contours—model contours predicted in the image plane, with image contours—contours extracted from the data. Therefore, we have to measure the discrepancy between a set of predictions (extremal contours) and a set of observations (image contours): we want to find the model's parameters that minimize this discrepancy.

Although the human body comprises several (five) kinematic chains, for the sake of clarity of exposition we consider only one such kinematic chain. We collect extremal-contour points from all the body-parts. Let $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_j, \dots, \boldsymbol{x}_m\}$ be the prediction vector, a set of m extremal-contour points. The components of this vector are 2-D points and they are parameterized by the kinematic- and free-motion parameter vector $\boldsymbol{\Phi}$, i.e. $\boldsymbol{x}(\boldsymbol{\Phi})$. Similarly, let $\mathcal{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_i, \dots, \boldsymbol{y}_k\}$ be the observation vector—a set of contour points observed in the image. In order to estimate the motion parameters one has to compare these two sets through a metric and to minimize it over the motion variables. Therefore, the problem can be generally stated as the minimization of a multi-variate scalar function E of (1).

There are several ways of defining and measuring the distance between two sets of points, $\mathcal{Y}$ and $\mathcal{X}$. One way of measuring this distance is to sum over one-to-one pairings $(\boldsymbol{x}_j, \boldsymbol{y}_i)$:

$$E(\mathcal{Y}, \mathcal{X}(\boldsymbol{\Phi})) = \sum_i \sum_j \alpha_{ij} \|\boldsymbol{y}_i - \boldsymbol{x}_j(\boldsymbol{\Phi})\|^2 \quad (56)$$

where the *hidden* variables α_{ij} are the entries of an association matrix: $\alpha_{ij} = 1$ if the observable $\boldsymbol{y}_i$ matches the prediction $\boldsymbol{x}_j$, and $\alpha_{ij} = 0$ otherwise. Therefore one has to solve both for the hidden variables and for the motion parameters (David et al. 2004).

5.1 The Hausdorf Distance

Another way of measuring the distance between two point-sets is to use the *Hausdorff distance* (Huttenlocher et al. 1993; Sim et al. 1999) which does not make use of explicit point pairings:

$$H(\mathcal{Y}, \mathcal{X}) = \max(h(\mathcal{Y}, \mathcal{X}), h(\mathcal{X}, \mathcal{Y})) \quad (57)$$

where $h()$ is called the *directed* Hausdorff distance: $h(\mathcal{Y}, \mathcal{X}) = \max_i(\min_j(\|\boldsymbol{y}_i - \boldsymbol{x}_j\|))$. The function $h()$ identifies the point in $\mathcal{Y}$ which is the farthest from any point in $\mathcal{X}$. The Hausdorff distance is the maximum between $h(\mathcal{Y}, \mathcal{X})$ and $h(\mathcal{X}, \mathcal{Y})$ and hence it measures the degree of mismatch between two point sets *without making explicit pairings* of points in one set with points in the other set. This means that many points of $\mathcal{Y}$ may be assigned to the same point of $\mathcal{X}$.

5.2 The Chamfer Distance

If the max operator in the Hausdorff distance is replaced by the summation operator, we obtain the normalized *directed* (or non-symmetric) chamfer distance:

$$DCD(\mathcal{Y}, \mathcal{X}) = \frac{1}{k}\sum_{i=1}^{k} \min_j(\|\boldsymbol{y}_i - \boldsymbol{x}_j\|). \quad (58)$$

The directed chamfer distance, or *DCD*, is a positive function and has the properties of identity and of triangle inequality but not of symmetry. It also has the desirable property that it can be computed very efficiently. Indeed, the *DCD* can be computed from the binary image of the observed image contour set $\mathcal{Y}$ using the *chamfer-distance image* $C_{\mathcal{Y}}$ (Borgefors 1986; Gavrila and Philomin 1999). The subscript $\mathcal{Y}$ reminds that this image is associated with the set $\mathcal{Y}$ of observed edge points. For each image site (pixel) with integer-valued image coordinates u_1 and u_2, the chamfer-distance image $C_{\mathcal{Y}}(u_1, u_2)$ returns the real-valued distance from this pixel to the nearest contour point of $\mathcal{Y}$. Therefore one can evaluate the distance from a predicted extremal-contour point $\boldsymbol{x} \in \mathcal{X}$ to its closest image contour by evaluating the chamfer-distance image at $\boldsymbol{x}$ with real-valued image coordinates x_1 and x_2.

We denote by $[x]$ the integer part of a real number x. Let $u_1 = [x_1]$ and $u_2 = [x_2]$ be the integer parts, and $r_1 = x_1 - [x_1]$ and $r_2 = x_2 - [x_2]$ be the fractional parts of the coordinates of a predicted point $\boldsymbol{x}$. The chamfer distance at $\boldsymbol{x}$ can be obtained by bi-linear interpolation of the chamfer-distance image:

$$\begin{aligned} D(\mathcal{Y}, \boldsymbol{x}) = {} & (1-r_1)(1-r_2)C_{\mathcal{Y}}(u_1, u_2) \\ & + r_1(1-r_2)C_{\mathcal{Y}}(u_1+1, u_2) \\ & + (1-r_1)r_2C_{\mathcal{Y}}(u_1, u_2+1) \\ & + r_1r_2C_{\mathcal{Y}}(u_1+1, u_2+1). \end{aligned} \quad (59)$$

5.3 Minimizing the Chamfer Distance

The minimization problem defined by (1) can now be written as the sum of squares of the chamfer distances over the predicted model contours:

$$f(\boldsymbol{\Phi}) = \frac{1}{2}\sum_{j=1}^{m} D_j^2(\mathcal{Y}, \boldsymbol{x}_j(\boldsymbol{\Phi})) = \frac{1}{2}\sum_{j=1}^{m} D_j^2(\boldsymbol{\Phi}). \tag{60}$$

In order to minimize this function over the motion parameters, we take its second-order Taylor expansion as well as the Gauss-Newton approximation of the Hessian:

$$f(\boldsymbol{\Phi}+\boldsymbol{d}) = f(\boldsymbol{\Phi}) + \boldsymbol{d}^\top \mathbf{J}_D^\top \boldsymbol{D} + \frac{1}{2}\boldsymbol{d}^\top \mathbf{J}_D^\top \mathbf{J}_D \boldsymbol{d} + \cdots$$

where $\boldsymbol{D}^\top = (D_1 \ \ldots \ D_m)$ and $\mathbf{J}_D^\top = [d\boldsymbol{D}/d\boldsymbol{\Phi}]^\top$ is the $n \times m$ matrix:

$$\mathbf{J}_D^\top = \begin{bmatrix} \dfrac{dD_1}{d\boldsymbol{\Phi}} & \cdots & \dfrac{dD_m}{d\boldsymbol{\Phi}} \end{bmatrix}. \tag{61}$$

The Chamfer-Distance Gradient The embedding of the tracker into such an optimization framework requires an analytic expression for the gradient of the error function to be minimized. The derivative of the chamfer distance D_j with respect to the motion parameters is the following matrix product:

$$\frac{dD_j}{d\boldsymbol{\Phi}} = \left(\frac{dD_j}{d\boldsymbol{x}}\right)^\top \frac{d\boldsymbol{x}}{d\boldsymbol{\Phi}}.$$

By noticing that $d[x]/dx = 0$, we immediately obtain an expression for $dD_j/d\boldsymbol{x}$:

$$\begin{aligned}\frac{\partial D_j}{dx_1} &= (1-r_2)(C_{\mathcal{Y}}(u_1+1,u_2) - C_{\mathcal{Y}}(u_1,u_2)) \\ &\quad + r_2(C_{\mathcal{Y}}(u_1+1,u_2+1) - C_{\mathcal{Y}}(u_1,u_2+1)), \\ \frac{\partial D_j}{dx_2} &= (r_1-1)(C_{\mathcal{Y}}(u_1+1,u_2) + C_{\mathcal{Y}}(u_1,u_2)) \\ &\quad + r_1(C_{\mathcal{Y}}(u_1+1,u_2+1) + C_{\mathcal{Y}}(u_1,u_2+1)).\end{aligned}$$

We recall that $d\boldsymbol{x}/d\boldsymbol{\Phi} = \mathbf{J}_I(\mathbf{A}+\mathbf{B})\mathbf{J}_H$ is the extremal-contour Jacobian defined in (19).

Issues related to the minimization of the chamfer distance can be found in (Knossow et al. 2006). Here we analyse the practical conditions under which this minimization should be carried out. At each time instant, the tracker is initialized with the previously found solution and (60) must be minimized. This minimization problem needs one necessary condition, namely that the $n \times n$ Hessian matrix has full rank. The Jacobian $\mathbf{J}_D$ is of size $m \times n$ and we recall that n is the number of variables to be estimated (the motion parameters) and m is the number of predictions (extremal contour points). To compute the inverse of $\mathbf{J}_D^\top \mathbf{J}_D$ we must have $m \geq n$ with n independent matrix rows.

5.4 How Many Cameras?

Since each prediction accounts for one row in the Jacobian matrix, one must somehow insure that there are n "independent" predictions. If each body part is viewed as a rigid object in motion, then it has six degrees of freedom. A set of three non-collinear points constrains these degrees of freedom. Whenever there are one-to-one model-point-to-image-point assignments, a set of three points is sufficient to constrain all six degrees of freedom. In the case of the chamfer distance there are no such one-to-one assignments and each model point yields only one constraint. Therefore, when one uses the chamfer distance, the problem is underconstrained since three non-collinear points yield three constraints only. Within a kinematic chain, the root body-part has six degrees of freedom and each body-part has $6+p$ degrees of freedom. Fortunately the body-parts are linked together to form kinematic chains. Therefore, one sensible hypothesis is to assume that the points at hand are evenly distributed among the body parts.

The kinematic human-body model that we use is composed of 5 kinematic chains that share a common root body-part, 19 body-parts, and 54 degrees of freedom (48 rotational joints and 6 free-motion parameters). Therefore, with an average of 3 points per body-part, there are in principle enough constraints to solve the tracking problem. Notice that the root-body part can arbitrarily be chosen and there is no evidence that one body-part is more suitable than another body-part to be the root part.

In practice there are other difficulties and problems. Due to total and/or partial occlusions, not all the body-parts can be predicted visible in one image. Therefore, it is impossible to insure that all the degrees of freedom are actually measured in one image. Even if a point attached to a visible body-part is predicted in the image, it may not be present in the data and/or it may be badly extracted and located. Non-relevant edges that lie in the neighborhood of a predicted location contribute to the chamfer distance and therefore complicate the task of the minimization process.

One way to increase the robustness of the tracker it to make recourse to redundant data. The latter may be obtained by using several cameras, each camera providing an independent chamfer distance error function. Provided that the cameras are *calibrated and synchronized* the method described above can be simultaneously applied to all the cameras. There will be several Jacobian matrices of the form of (61) (one for each camera) and these matrices can be combined together into a unique Jacobian, provided that a common world reference frame is being used (Martin and

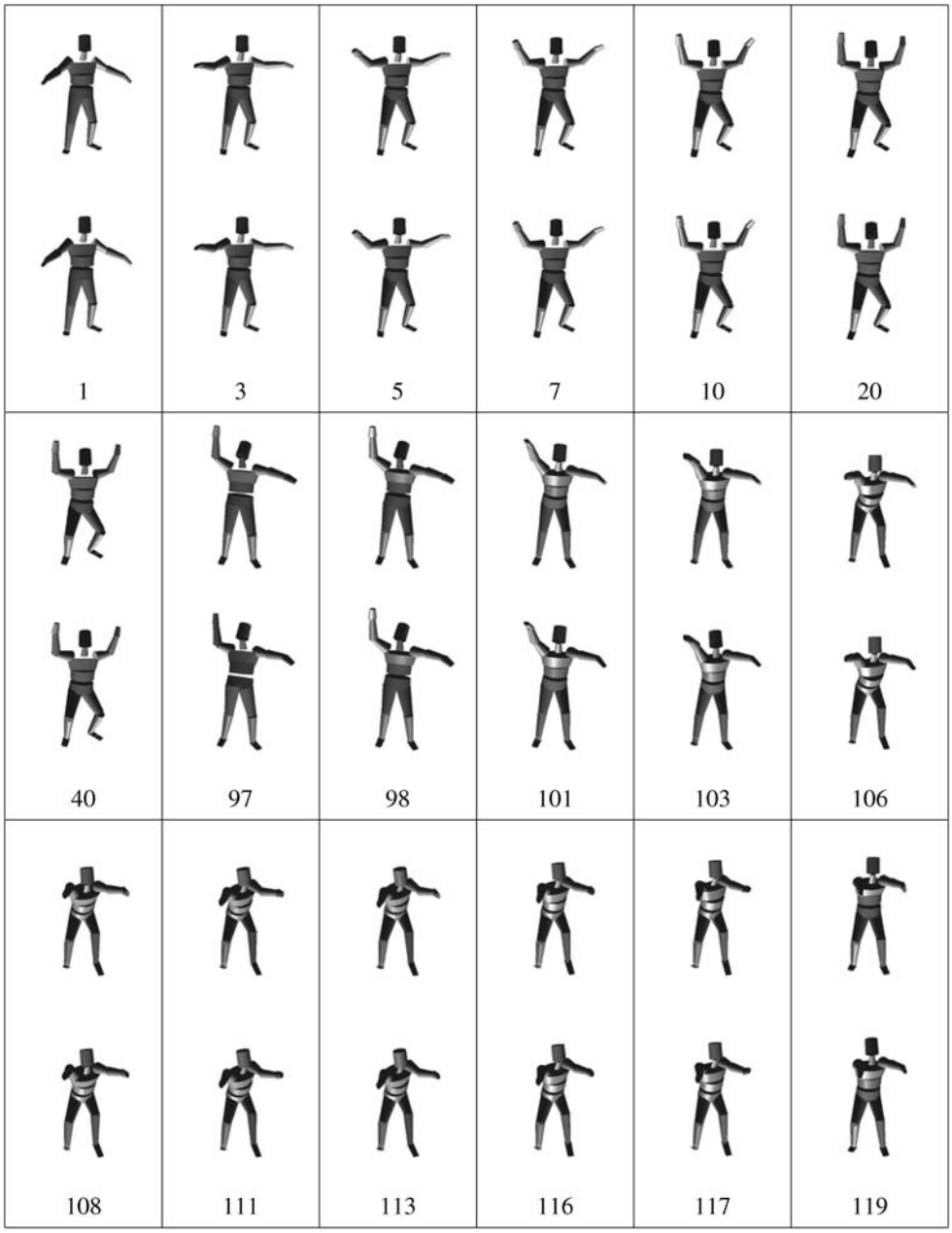


Fig. 4 This figure compares the ground truth (*first, third and fifth rows*) with the estimated poses (*second, fourth and sixth rows*). The ground-truth poses were used to simulate silhouette data to be used by the tracker

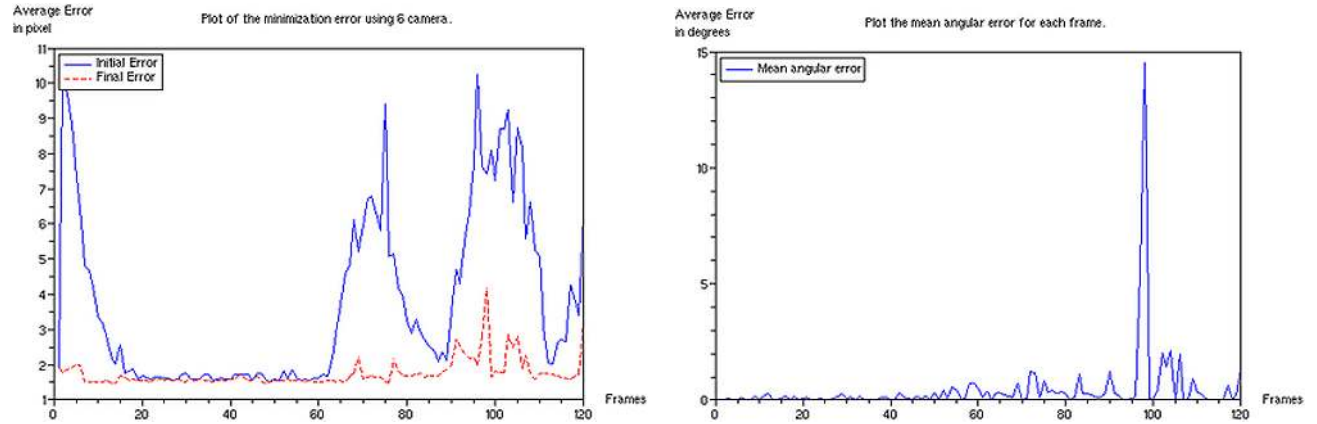
Horaud 2002). Therefore, by using several cameras one increases the number of predictions (columns in the Jacobian) without increasing the number of variables.

It is worthwhile to notice that the extremal contours viewed with one camera are different than the extremal contours viewed with another camera. Indeed, these two sets of contours correspond to different physical points onto the surface. One great advantage of using extremal contours in order to fit the model parameters with the data is that there is no need to establish matches across images taken with distinct cameras.

6 Experiments with Simulated and Real Data

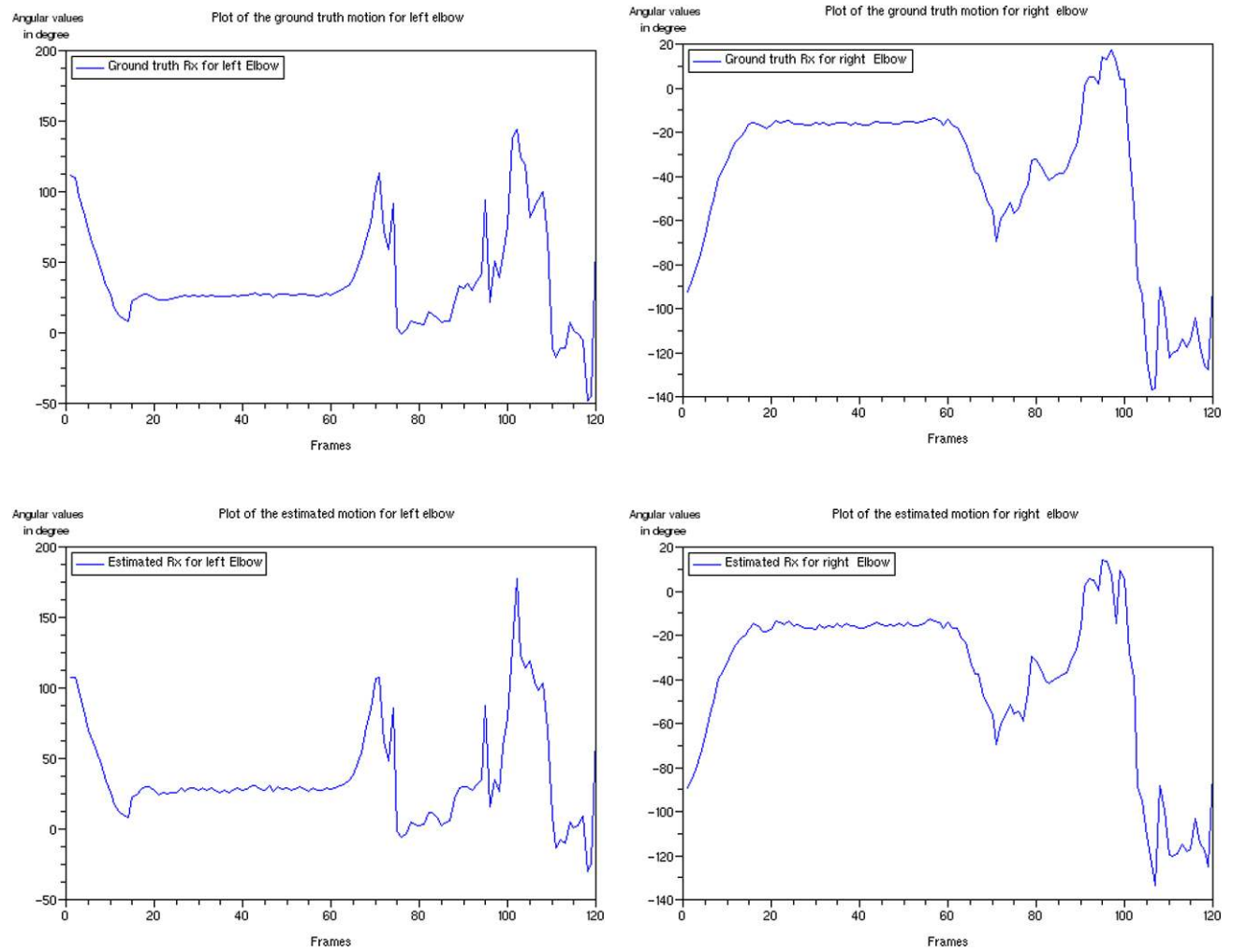
The simulated data were produced using a motion capture system and an animation software. This outputs trajectories for the motion parameters of our human-body model. From these trajectories we generated a sequence of model motions. From each pose of the model we computed extremal contours for six images associated with six virtual cameras. We simulated a total of 120 frames for each image sequence.

Next, we applied our method to these contours. Figure 4 shows the simulated poses (top rows) as well as the esti-



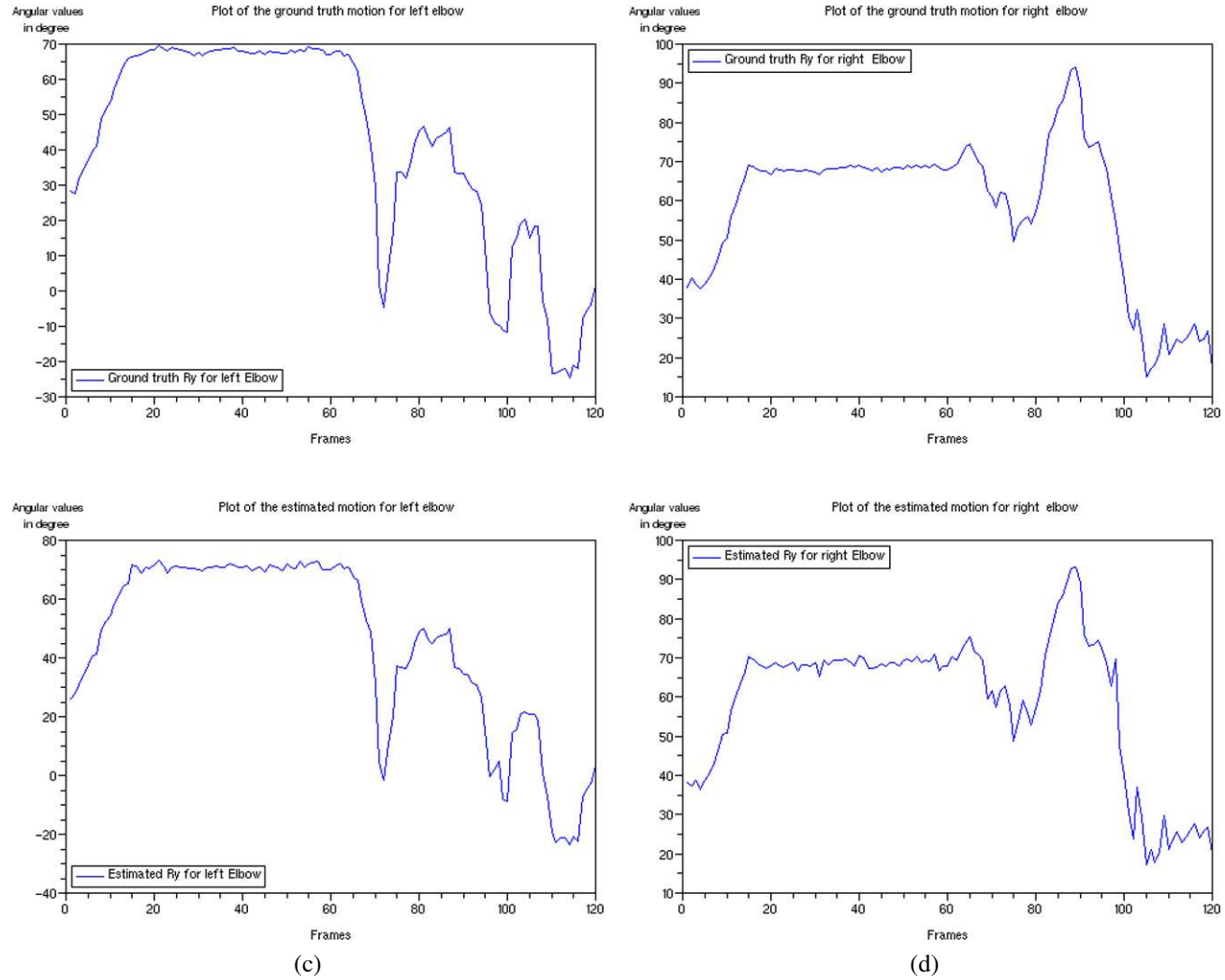
(a) (b)

Fig. 5 **a** The error between the image contours and the projected extremal contours before minimization (*top curve*) and after minimization (*bottom curve*). **b** The average error between the simulated motion parameters and the estimated ones



(a) (b)

Fig. 6 Ground-truth and estimated joint-angle trajectories for the left and right elbows



(c) (d)

Fig. 6 (*continued*)

mated poses (bottom rows). Figures 5, 6, and 7 compare the results of our method with the ground truth. Figure 5-b plots the average error between the true motion parameters and the estimated ones. One may notice that the average error remains within 2^0, with the exception of frame 98 for which the average error is 15^0: this error is due to 180^0 ambiguity associated with one of the joints. Nevertheless, the tracker was able to recover from this large error. Figure 5a shows the initial error between the predicted contours and the image contours (top curve) as well as the error once the motion parameters were fitted using the minimization described in Sect. 5.3. The failure of the tracker at frame 98 corresponds to an error of 4–5 pixels. The initial mean error is 3.7 pixels whereas the final mean error is 1.5 pixels.

In more detail, Figs. 6 and 7 compare the estimated trajectories with the ground truth for the left and right elbows and for the left and right shoulders. Both the elbow and shoulder are modeled with two rotational degrees of freedom.

In order to track real human-body motions we used a setup composed of six cameras that were accurately calibrated and whose video outputs are finely synchronized. Fine synchronization (of the order of 10^{-6} s) and fast shutter speed (10^{-3} s) allow one to cope with fast motions. The camera setup is shown on Fig. 8. It consists in six Firewire cameras that deliver 600×800 uncompressed images at 30 frames per second.

We used two different persons, Ben and Erwan. These two persons have the same size and therefore we used the same roughly estimated parameters for the elliptical cones modeling the body parts.

We gathered three sets of data shown on Fig. 1 (Erwan-1), Fig. 11 (Ben) and Fig. 12 (Erwan-2). For each data set, the figures show the images associated with the first three cameras, the associated silhouettes, and the estimated pose of the model displayed from the viewpoint of the third camera. The extremal contours eventually fitted to the data are shown overlayed onto the raw images.

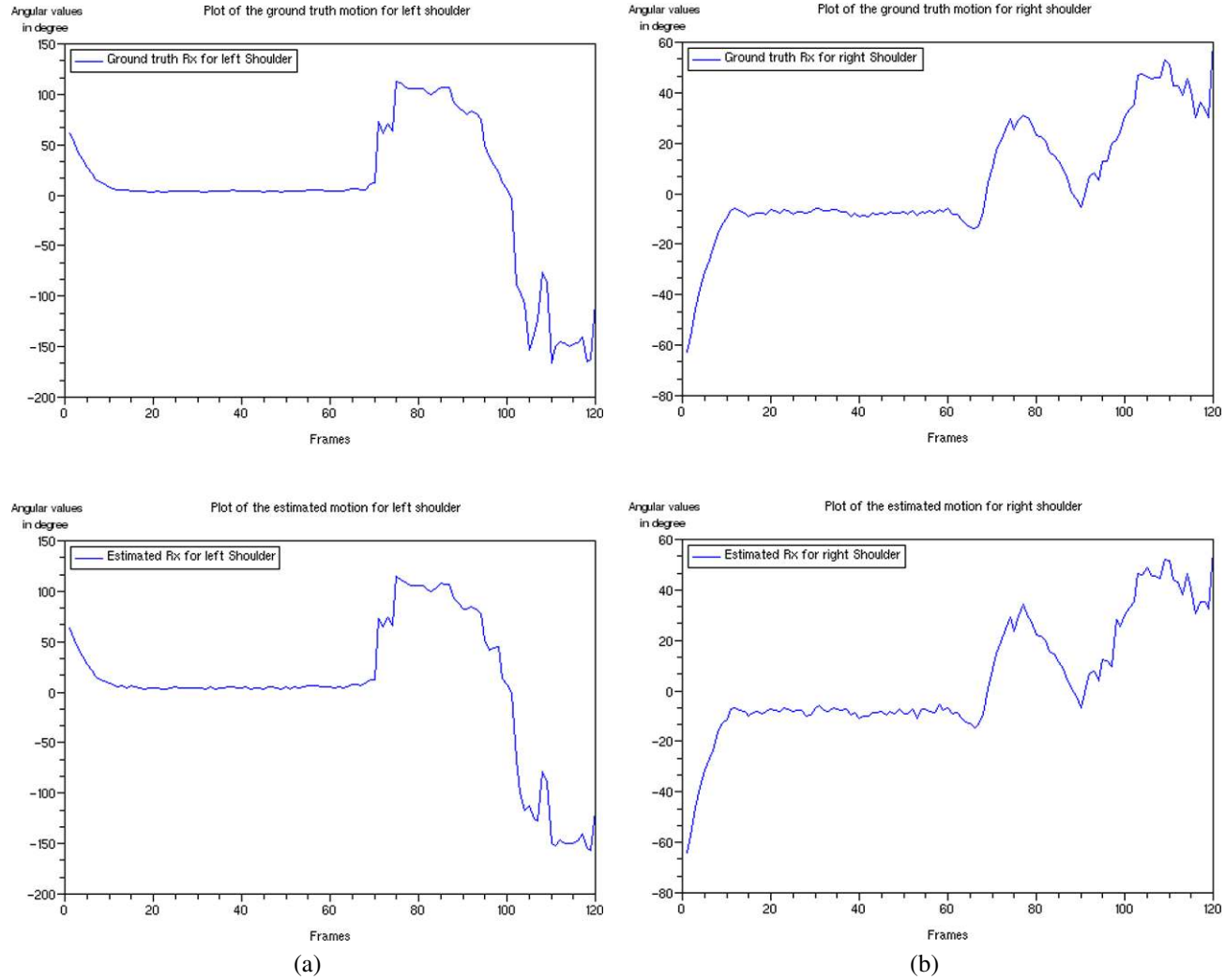


Fig. 7 Ground-truth and estimated joint-angle trajectories for the left and right shoulders

The tracking is initialized by incremental pose estimation of the body parts. We start with an initial guess (Fig. 9a) from which the pose of the root body part is first estimated (Fig. 9b). This is followed by the pose estimation of other body parts (Fig. 9c). The final kinematic pose found by this initialization process is shown on Fig. 9d. The first example, Erwan-1, has 250 frames, the second example, Ben, has 800 frames and the third example, Erwan-2, has 200 frames. Notice that the Erwan motions involve all the degrees of freedom of the articulated model, as well as the motion of the root body-part.

The efficiency of minimization-based trackers, as the one described here resides in number of iterations needed by the optimization algorithm to converge. In all the examples described in this paper we minimized an error function that has two terms: one term corresponds to the rigid motions of the body parts and the other terms corresponds to the sliding of the contour generator on the body-parts' surface. Under these circumstances, the tracker converges in 3 to 5 iterations and the RMS image error is, in this case, 1.5 pixels, Fig. 13 (bold plots). If the sliding-motion term is left out the efficiency of the tracker is substantially degraded because the optimizer needs twice more iterations for an RMS error of 2.3 pixels, Fig. 13 (dashed plots).

It is worthwhile to notice that we used a human model with the same measurements for the two persons (body-part parameters such as the size of the arms, feet, thighs, head, torso, etc.). More accurate model parameters (finely adjusted to each person) will result in smaller RMS errors.

6.1 Comparison with Marker-Based Motion Capture Data

One way to quantitatively evaluate the performance of markerless human tracking methods such as the one described in this paper, is to compare it with a marker-based motion capture system. Until recently it was believed that markerless motion capture systems cannot compete with marker-

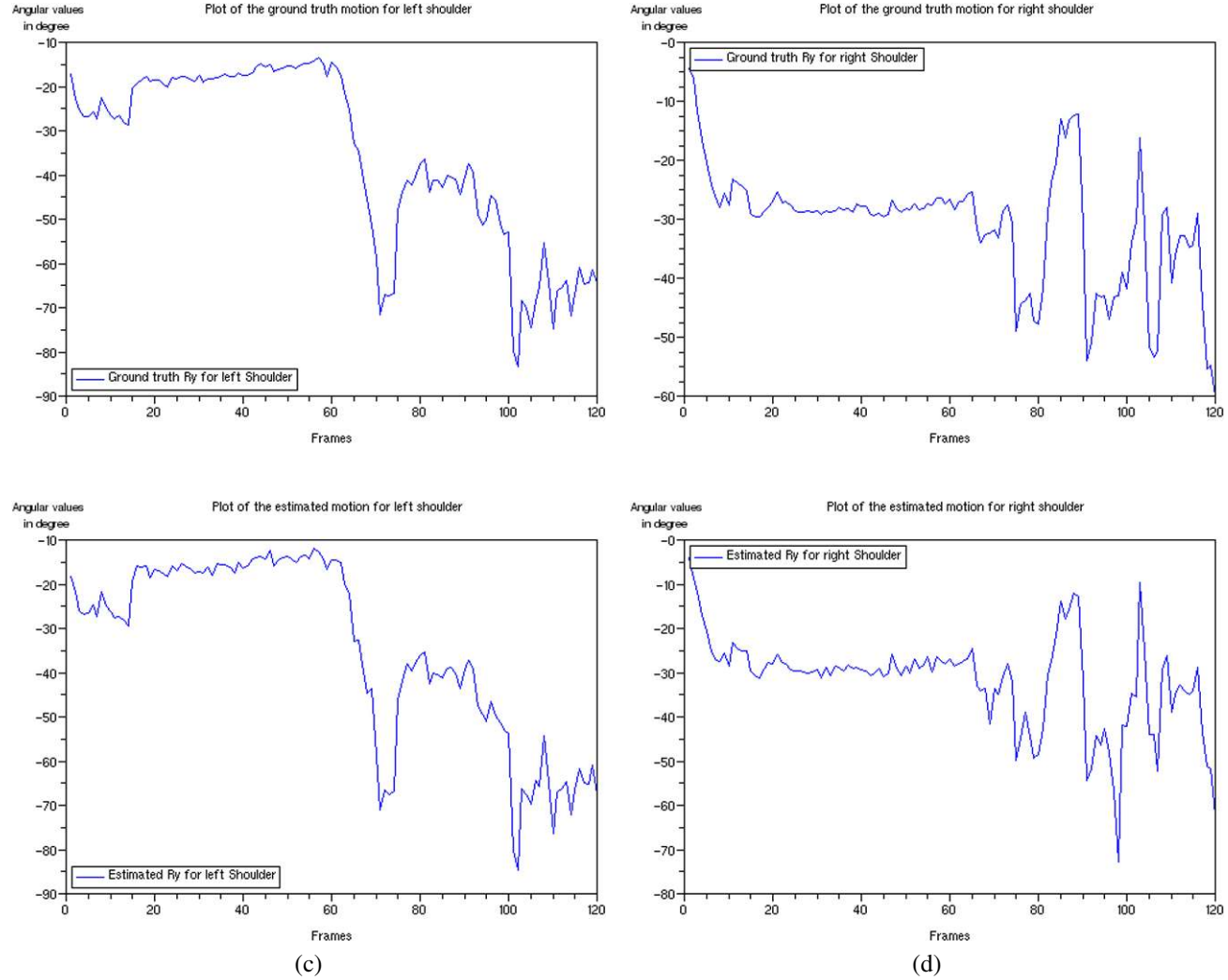


Fig. 7 (*continued*)

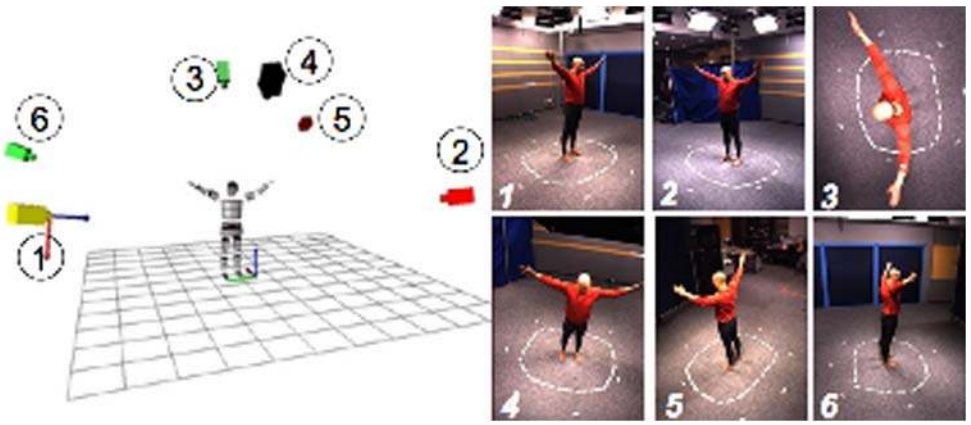


Fig. 8 The camera setup used in our experiments. The cameras are calibrated with respect to a global reference frame. The human body model is shown in it's reference position

Fig. 9 Initialization: Starting from an initial guess, (**a**), the pose of the root body-part is first estimated, (**b**), followed by the incremental estimation of the remaining body parts, (**c**) and (**d**)

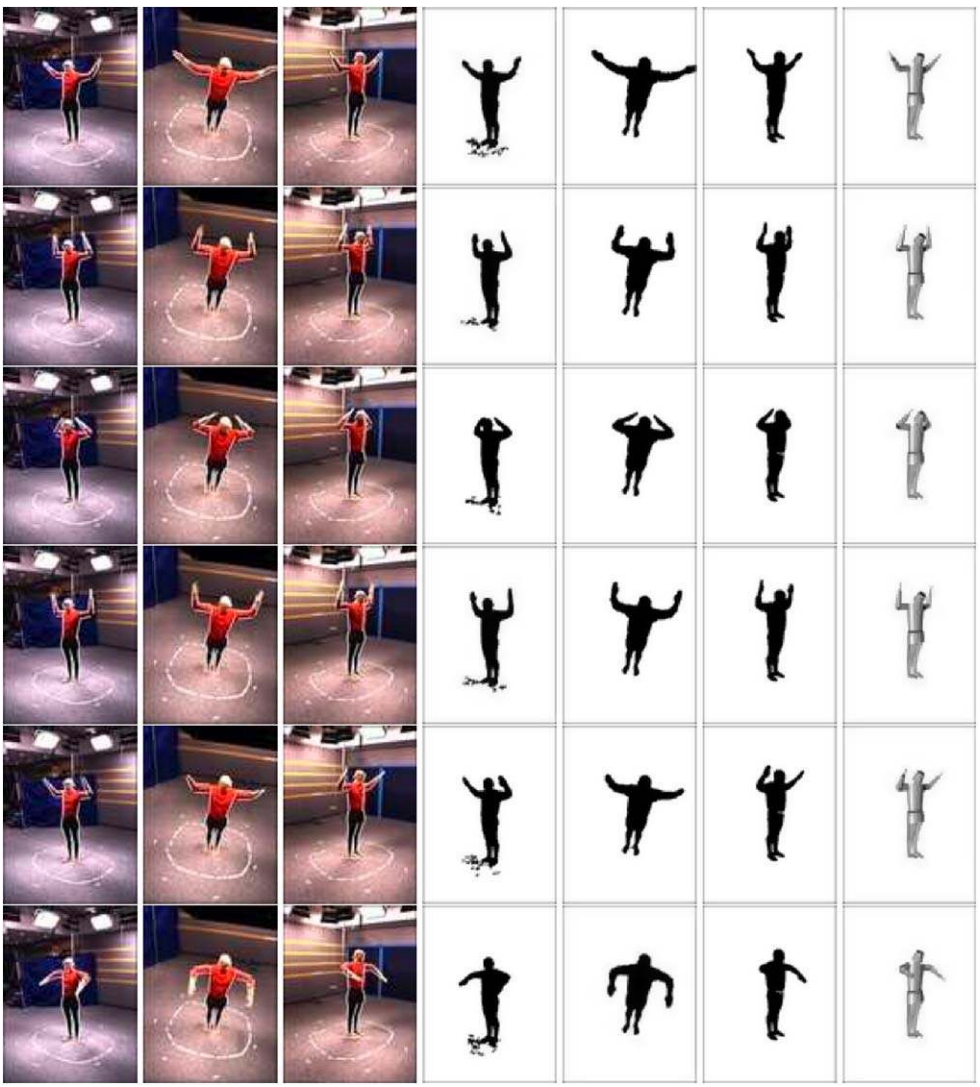


Fig. 10 The result of tracking Ben with 6 cameras over 800 frames (continues on the next figure)

based systems (Gleicher and Ferrier 2002). We performed the following experiment and evaluation which is similar in spirit with the work described in (Balan et al. 2005; Sigal and Black 2006).

We equipped a room with two camera systems, our 6-camera system and a VICON system using 8 cameras. We simultaneously gathered markerless and marker-based data with these two systems. While our system gathers 8-

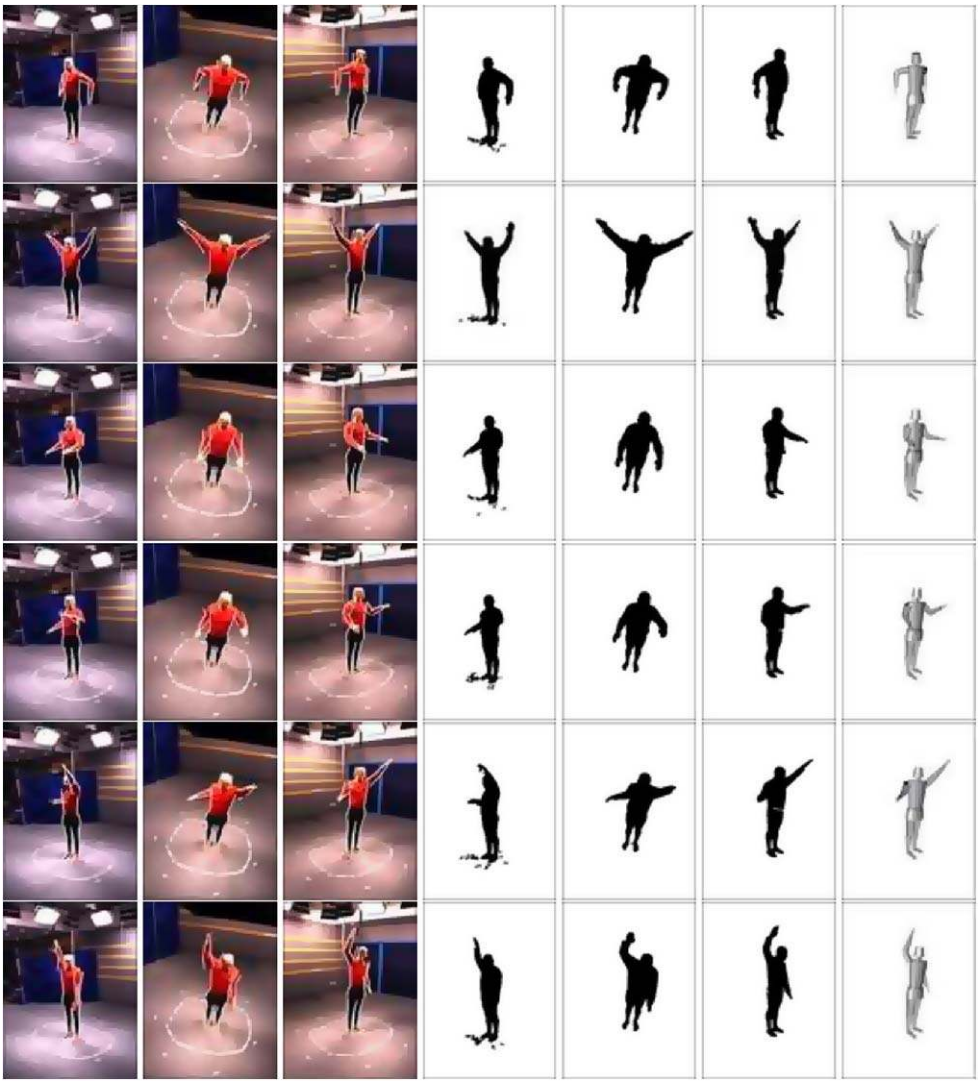


Fig. 11 The result of tracking Ben with 6 cameras over 800 frames (continued from the previous figure)

bit color images, the VICON system only gathers the image locations of the markers. Figure 14 shows two out of the six image sequences (first and second columns), the corresponding articulated poses found with our method (the third and fourth columns show the pose of the model from the viewpoints of the two cameras shown onto the left side of the figure). Both our algorithm and an algorithm based on the VICON data output joint trajectories. These trajectories are used to estimate the poses of a virtual character, as shown on the fifth and sixth columns. Character animation using both these types of data are available at http://perception.inrialpes.fr/~Knossow.

7 Conclusion

In this paper we proposed a contour-based method for tracking the motion of articulated objects. The main contributions of the paper are as follows. We derived an exact kinematic parameterization of the extremal contours produced by developable surfaces. We combined this parameterization with the zero-reference kinematic model that is well suited for representing the space of human motions, i.e., a combination of both the space of articulated motions (spanned by rotational joints) and the space of free motions (spanned by three rotations and three translations). We derived an analytical expression for the Jacobian linking joint- and free-motion velocities to extremal-contour velocities and we showed how this Jacobian matrix can be plugged into a non-linear minimization method. We made explicit two components of the Jacobian: a rigid-motion component and a sliding-motion component. The cost function uses the directed chamfer distance between extremal contours predicted by the model and image contours extracted from silhouettes. One major advantage of using the directed chamfer

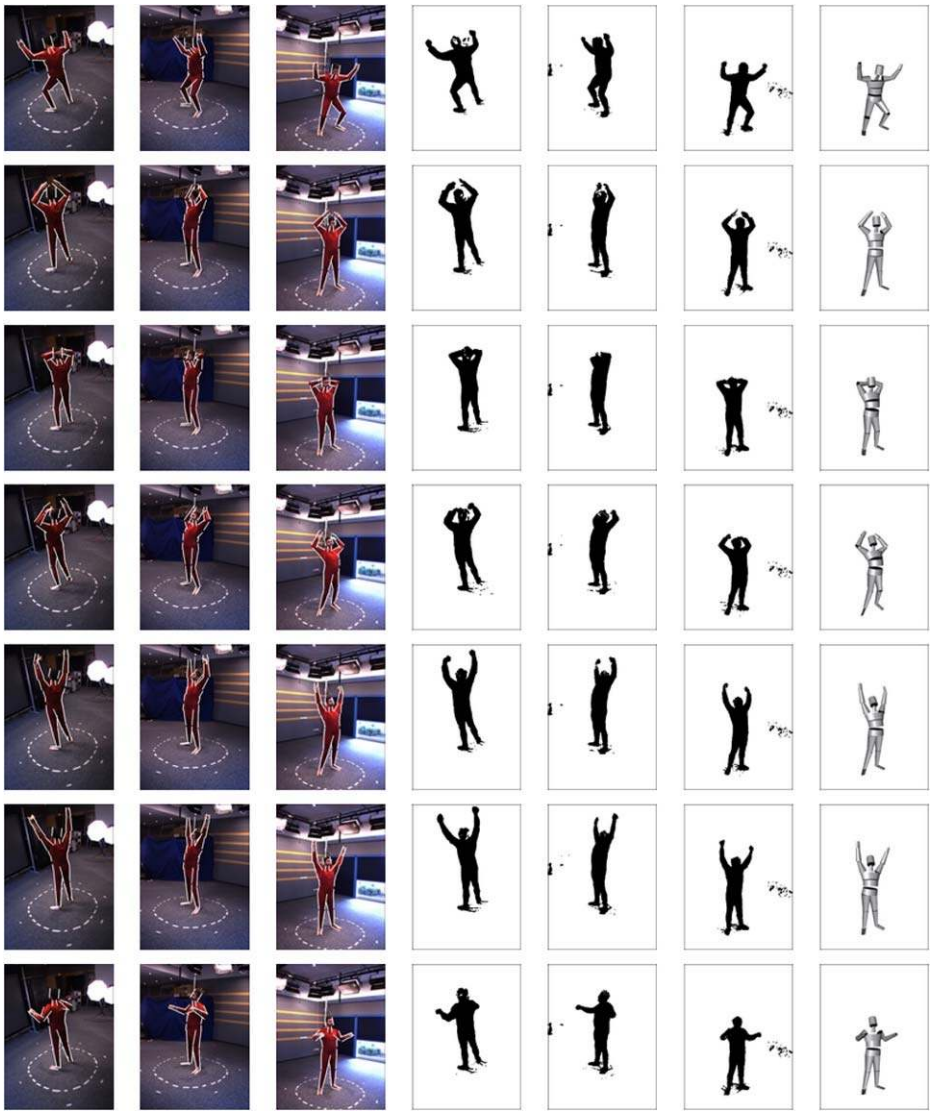


Fig. 12 The result of tracking Erwan-2 with 6 cameras over 250 frames

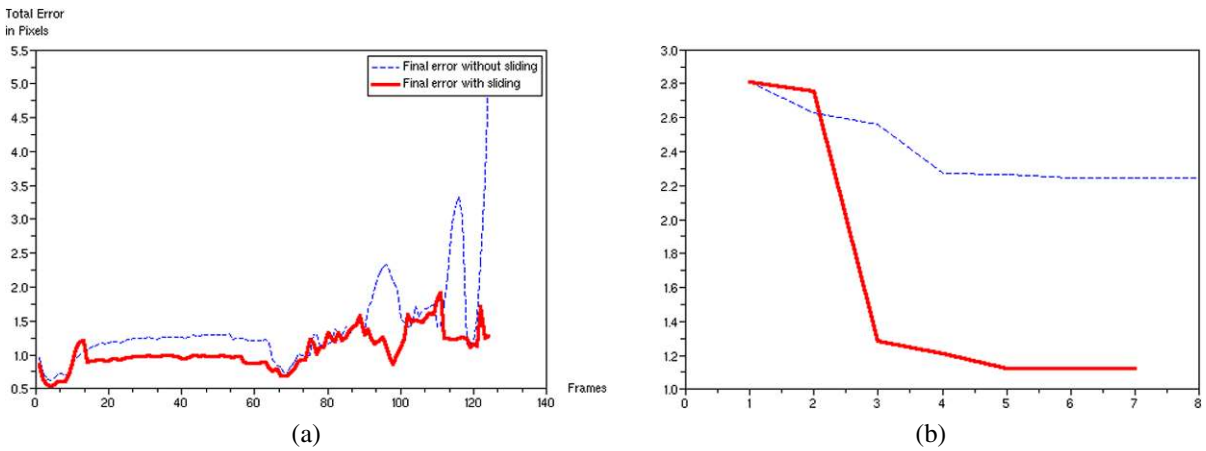


Fig. 13 These graphs show the advantage of using the sliding motion. **a** A comparison between the minimization error obtained using the sliding motion (*bold*) and without using it (*dashed*). **b** A comparison of the speed of convergence of the minimization method (number of iterations) with the sliding-motion component (*bold*) and without using it (*dashed*)

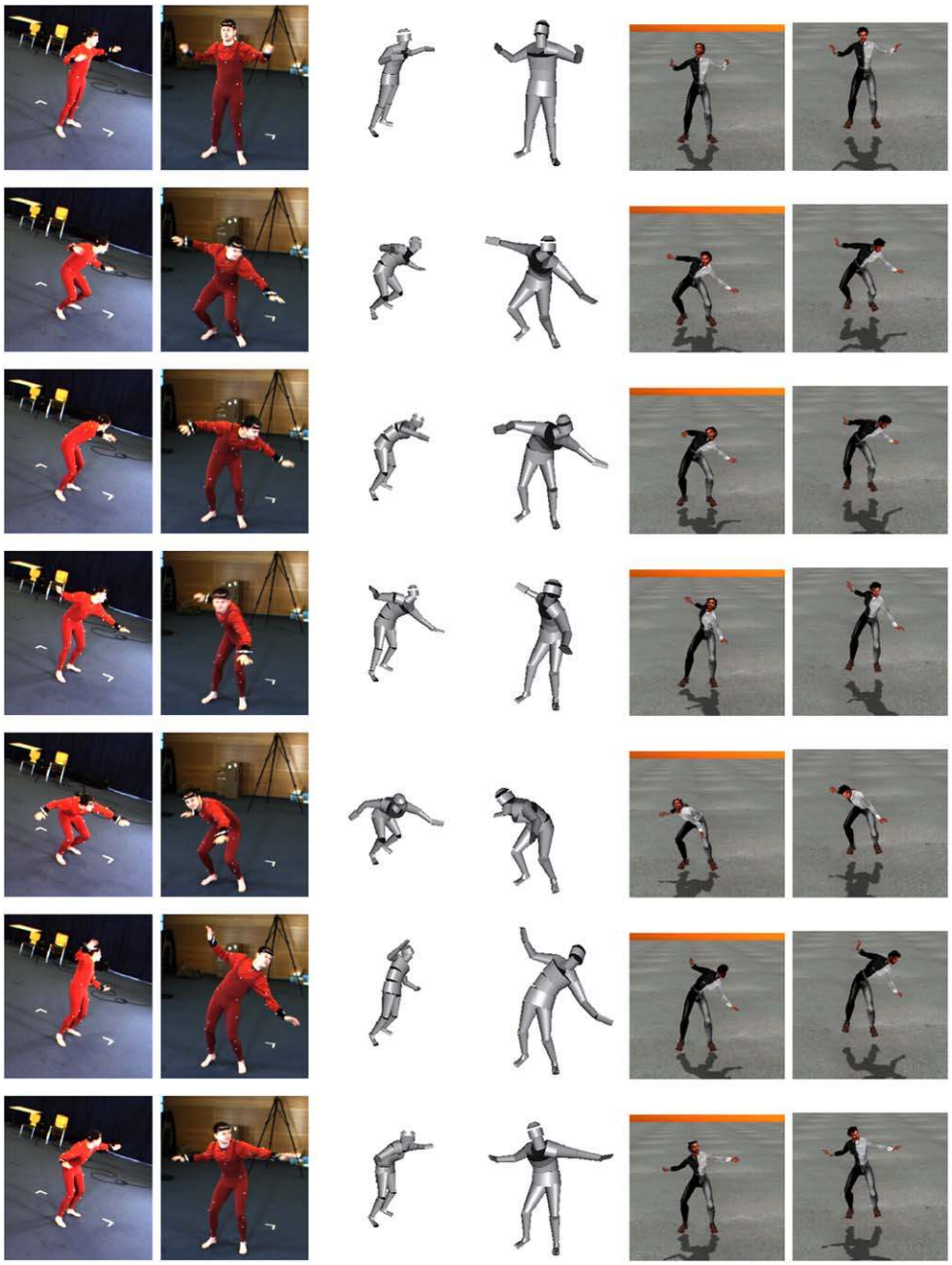


Fig. 14 This figure shows a *qualitative* comparison between markerless and marker-based motion capture. The videos shown on to the *left* (together with four other videos which are not shown) produced the results shown on the *third, fourth and fifth columns*. The *last column* shows the result obtained from a method that uses a 8-camera VICON system to locate image markers

distance is that it does not require one-to-one assignments between image observations and model features.

Moreover, we analysed the conditions under which the minimization process can be carried out effectively, i.e., without failures due to numerical instabilities. Although, in principle, one camera may suffice, in practice it is desirable to have several images gathered simultaneously with several cameras. We carried out a large number of experiments with both simulated and real data. The tracker performed very well and is able to recover from badly estimated poses. We performed experiments with both simulated and real data gathered with six cameras. We compared the angle trajectories obtained with our method with trajectories obtained using a marker-based commercial system that uses eight cameras. We plan to compare more thoroughly our method with other methods within formal evaluation protocols.

In the future we plan to have a probabilistic look at the problem while maintaining the deterministic relation-

ship between extremal contours and moving articulated objects. One possibility is to consider the graphical model framework successfully applied to the pictorial recognition of articulated objects (Felzenswalb and Huttenlocher 2005), (Ronfard et al. 2002). Another possibility is to view each extremal contour as a thin and elongated cluster and to apply model-based clustering methods (Fraley and Raftery 2002). Both these approaches raise the problem of relating the parameters of the probability distribution function at hand with the kinematic parameterization proposed in this paper. It is worthwhile to notice that the clustering framework just mentioned is consistent with the chamfer distance which can be modified such that it accounts for a probabilistic association between a set of observations and a model.

References

Agarwal, A., & Triggs, W. (2006). Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *28*(1), 44–58.

Balan, A. O., Sigal, L., & Black, M. J. (2005). A quantitative evaluation of video-based 3D person tracking. In *PETS'05* (pp. 349–356).

Barrow, H. G., & Tenenbaum, J. M. (1981). Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, *17*(1–3), 75–116.

Borgefors, G. (1986). Distance transformation in digital images. *Computer Vision, Graphics, and Image Processing*, *34*(3), 344–371.

Bregler, C., Malik, J., & Pullen, K. (2004). Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, *56*(3), 179–194.

Cheung, K. M., Baker, S., & Kanade, T. (2005a). Shape-from-silhouette across time, part I: theory and algorithms. *International Journal of Computer Vision*, *62*(3), 221–247.

Cheung, K. M., Baker, S., & Kanade, T. (2005b). Shape-from-silhouette across time, part II: applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, *63*(3), 225–245.

David, P., DeMenthon, D. F., Duraiswami, R., & Samet, H. (2004). Softposit: simultaneous pose and correspondence determination. *International Journal of Computer Vision*, *59*(3), 259–284.

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

Deutscher, J., Blake, A., & Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *Computer vision and pattern recognition* (pp. 2126–2133).

Do Carmo, M. P. (1976). *Differential geometry of curves and surfaces*. New York: Prentice-Hall.

Drummond, T., & Cipolla, R. (2001). Real-time tracking of highly articulated structures in the presence of noisy measurements. In *ICCV* (pp. 315–320).

Felzenswalb, P., & Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*(1), 55–79.

Forsyth, D. A., & Ponce, J. (2003). *Computer vision—a modern approach*. New Jersey: Prentice Hall.

Forsyth, D. A., Arikan, O., Ikemoto, L., O'Brien, J., & Ramanan, D. (2006). Computational studies of human motion, part 1: tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision*, *1*(2), 77–254.

Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, *97*, 611–631.

Gavrila, D. M. (1999). The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, *73*(1), 82–98.

Gavrila, D. M., & Davis, L. S. (1996). 3D model-based tracking of humans in action: a multi-view approach. In *Conference on computer vision and pattern recognition* (pp. 73–80), San Francisco, CA.

Gavrila, D. M., & Philomin, V. (1999). Real-time object detection for smart vehicles. In *IEEE Proceedings of the seventh international conference on computer vision* (pp. 87–93), Kerkyra, Greece.

Gleicher, G., & Ferrier, N. (2002). Evaluating video-based motion capture. In *Proceedings of the computer animation 2002* (pp. 75–80), Geneva, Switzerland, June 2002.

Huttenlocher, D. P., Klanderman, G. A., & Rucklidge, W. J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(9), 850–863.

Kakadiaris, I., & Metaxas, D. (2000). Model-based estimation of 3D human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(12), 1453–1459.

Kehl, R., & Van Gool, L. J. (2006). Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*, *103*(23), 190–209.

Knossow, D., Ronfard, R., Horaud, R., & Devernay, F. (2006). Tracking with the kinematics of extremal contours. In *Lecture notes in computer science*. *Computer vision—ACCV 2006* (pp. 664–673), Hyderabad, India, January 2006. Berlin: Springer.

Koenderink, J. (1990). *Solid shape*. Cambridge: The MIT Press.

Kreyzig, E. (1991). *Differential geometry*. New York: Dover. Reprint of a U. of Toronto 1963 edition.

Martin, F., & Horaud, R. (2002). Multiple camera tracking of rigid objects. *International Journal of Robotics Research*, *21*(2), 97–113.

McCarthy, J. M. (1990). *Introduction to theoretical kinematics*. Cambridge: MIT Press.

Mikic, I., Trivedi, M. M., Hunter, E., & Cosman, P. C. (2003). Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, *53*(3), 199–223.

Moeslund, T. B., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, *104*(2), 90–126.

Mooring, B. W., Roth, Z. S., & Driels, M. R. (1991). *Fundamentals of manipulator calibration*. New York: Wiley.

Murray, R. M., Li, Z., & Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. Ann Arbor: CRC Press.

Plaenkers, R., & Fua, P. (2003). Articulated soft objects for multi-view shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(10), 1182–1187.

Ronfard, R., Schmid, C., & Triggs, W. (2002). Learning to parse pictures of people. In *Proceedings of the 7th European conference on computer vision* (Vol. 4, pp. 700–714), Copenhagen, Denmark, June 2002. Berlin: Springer.

Sigal, L., & Black, M. J. (2006). *Humaneva: synchronized video and motion capture dataset for evaluation of articulated human motion* (Technical Report CS-06-08). Department of Computer Science, Brown University, Providence, RI 02912, September 2006.

Sim, D. G., Kwon, O. K., & Park, R. H. (1999). Object matching algorithms using robust Hausdorff distance measures. *IEEE Transactions on Image Processing*, *8*(3), 425–429.

Sminchisescu, C., & Triggs, W. (2003). Kinematic jump processes for monocular 3D human tracking. In *International conference on computer vision and pattern recognition* (Vol. I, pp. 69–76), June 2003.

Sminchisescu, C., & Triggs, W. (2005). Building roadmaps of minima and transitions in visual models. *International Journal of Computer Vision*, *61*(1), 81–101.

Song, Y., Goncalves, L., & Perona, P. (2003). Unsupervised learning of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(7), 814–827.

Toyama, K., & Blake, A. (2002). Probabilistic tracking with exemplars in a metric space. *International Journal of Computer Vision*, *48*(1), 9–19.