Tracking with the kinematics of extremal contours

David Knossow, Rémi Ronfard, Radu Horaud, and Frédéric Devernay

INRIA Rhone-Alpes, 655 Av. de l'Europe, 38330 Montbonnot, France

Abstract. *This paper addresses the problem of articulated motion tracking from image sequences. We describe a method that relies on an explicit parameterization of the extremal contours in terms of the joint parameters of an associated kinematic model. The latter allows us to predict the extremal contours from the body-part primitives of an articulated model and to compare them with observed image contours. The error function that measures the discrepancy between observed contours and predicted contours is minimized using an analytical expression of the Jacobian that maps joint velocities onto contour velocities. In practice we model people both by their geometry (truncated elliptical cones) and with their articulated structure – a kinematic model with 40 rotational degrees of freedom. We observe image data gathered with several synchronized cameras. The tracker has been successfully applied to image sequences gathered at 30 frames/second.*

1 Introduction and background

In this paper we address the problem of tracking complex articulated motions, such as human motion, from visual data. More precisely, we describe humans by a set of kinematically-articulated body parts with smooth surfaces. These surfaces project onto images as extremal contours. We observe humans with several cameras, we extract image contours and we estimate the motion parameters by minimizing the discrepancy between predicted extremal contours and image contours.

The problem of human motion recovery has been thoroughly studied in the recent past using either one or several cameras and without artificial markers [1]. Previous work may be classified into two main approaches.

One approach extracts image features that can be used in the same way as markers, such as texture [2] or point features [3]. Those methods can be implemented in a straightforward manner since they have an explicit differential model of the kinematics, and the latter can be inverted using non-linear least squares methods. The difficulty is then to relate the positions of the features with a geometric model of the human body. In practice, this usually implies full knowledge of both the geometry and the appearance of the human actor [4],

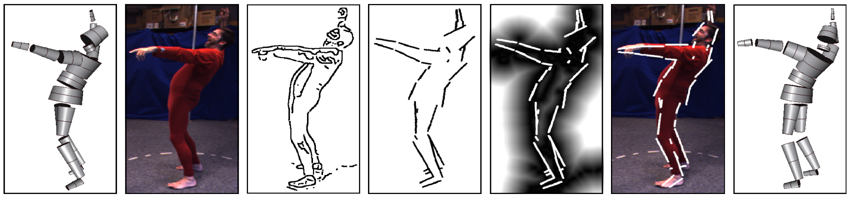

Fig. 1. From left to right : The current model is matched against a new image. The contours extracted from this image are compared with the extremal contours predicted from the model using the chamfer-distance image. Finally, the newly estimated model is consistent with this image.

although recent advances in multi-body factorization may provide solutions for simultaneously recovering the motion *and* the structure [5].

Another approach relies on contours [6] or on silhouettes [7–9]. It is possible to relate the deformation of a 2-D (image) silhouette to the geometry and the motion of the articulated object which generated that silhouette. Methods based on deformable silhouettes [10] can cope only with limited changes in viewpoint and pose, and cannot deal with occlusions between primitives. Statistical methods in general and regressive models in particular are used to relate the shape of a silhouette with three-dimensional motion in a lower-dimensional motion space, learned from examples of a specific activity [11].

A slightly different approach was taken in [12], [13] for tracking mechanical parts with sharp edges. By parameterizing the allowable contour deformations with the actual degrees of freedom of the underlying rigid motions of the parts, they demonstrated increased robustness and efficiency over fully deformable active contours for tracking such objects. In the case of human motion tracking, the task is made harder by the fact that the human body has fewer sharp edges (if none), and its silhouette stems from the projection of smooth surfaces rather than surfaces with sharp edges.

Problem formulation and originality. We model articulated objects such as humans using *truncated elliptical cones* as basic primitives. These primitives are joined together to form an articulated structure. Each joint has one to three rotational degrees of freedom: let $\boldsymbol{\Phi}$ be an n-dimensional vector whose components are the motion parameters – the joint angles. The smooth surface of a primitive projects onto an image as an *extremal contour*. The apparent motion of this contour is a function of both the motion of the primitive and the motion of the *contour generator* lying onto the smooth surface. An important contribution of this work is to establish the relationship between the joint-angle velocities, $\dot{\boldsymbol{\Phi}} = \partial \boldsymbol{\Phi} / \partial t$, and the image velocity of a point lying onto an extremal contour, $\boldsymbol{v}$:

$$\boldsymbol{v} = \mathbf{J} \dot{\boldsymbol{\Phi}} \tag{1}$$

Matrix $\mathbf{J}$ will be referred to as the *extremal contour Jacobian*. The analytic expression of this Jacobian allows us to cast the tracking problem into a non-linear optimization problem. Therefore, the problem of articulated-motion tracking will be formulated as the problem of minimizing a distance function between sets of image contours (gathered simultenously from several cameras) and sets of extremal contours. This can be written as:

$$\min_{\boldsymbol{\Phi}} E(\mathcal{Y}, \mathcal{X}(\boldsymbol{\Phi})) \tag{2}$$

where E is an error or a distance function, $\mathcal{Y}$ is the set of observed image contours and $\mathcal{X}(\boldsymbol{\Phi})$ is the set of predicted extremal contours. There are several ways of computing the distance between image and model contours, including the sum over point-to-point distances, the Hausdorff distance, and so forth. We use the chamfer distance and has several interesting features. It does not require model-contour-to-image contour matches and its computation is fast. Moreover, we treat the chamfer distance as a differentiable function. In practice, a chamfer-distance image is computed from the data. It combines image edges with a binary silhouette which acts both as a mask and as a way to suppress artifacts in the chamfer-distance image.

Paper organization. The remainder of this paper is organized as follows. In section 2 we derive an analytical solution that relates the motion of an extremal contour to joint parameters of an articulated object. In section 3 we provide an explicit expression for measuring the distance between image contours and extremal contours; Moreover, we explain the advantages of using both edges and silhouettes. Finally, we present examples with complex and realistic motions that require several cameras (section 4).

2 Kinematics of extremal contours

As we already explained above, we use truncated elliptical cones as our basic primitives, i.e., Figure 2. These primitives are linked together with rotational joints (with one, two, or three degrees of freedom) to form a kinematic chain. Therefore, the motion of each such primitive is a constrained motion. Let $\mathbf{R}$ and $\boldsymbol{t}$ denote the rotation and translation of a primitive-centered frame with respect to a world-centered frame. Both $\mathbf{R}$ and $\boldsymbol{t}$ are therefore parameterized by the joint angles $\boldsymbol{\Phi} = (\phi_1, \ldots, \phi_n)$, i.e., we have $\mathbf{R}(\boldsymbol{\Phi})$ and $\boldsymbol{t}(\boldsymbol{\Phi})$.

Moreover we consider the smooth surface of the elliptical cone. This surface is present in the image under the form of extremal contours. The image motion of a point belonging to such an extremal contour should, therefore, depend on the kinematic motion of the corresponding cone. One can further define a *contour generator* onto the cones's smooth surface – the locus of points where the surface is tangent to lines of sight. When the cone moves, the contour generator moves as well and is constrained both by the kinematic motion of the cone itself and by the relative position of the cone with respect to the camera. Therefore, the contour generator has two motion components and we must explicitly estimate

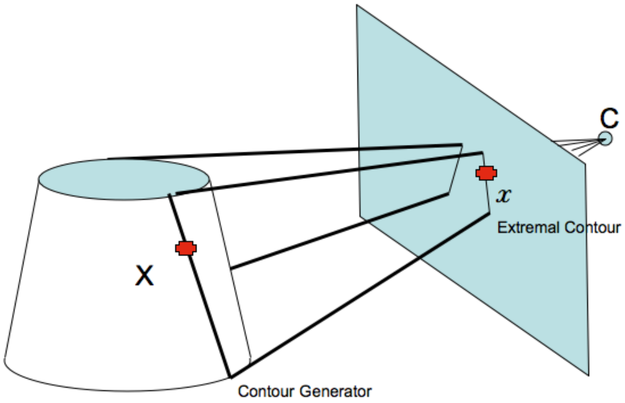

Fig. 2. A truncated elliptical cone projects onto an image as a pair of *extremal contours*. The 2-D motion of these extremal contours is a function of both the motion of the cone and the sliding of the *contour generator* along the smooth surface of the cone.

these components. First, we will develop an analytical solution for computing the contour generator as a function of the motion parameters. The extremal contour is simply the projection of the contour generator. Second, we will develop an expression for the image Jacobian that maps joint-velocities onto image point-velocities.

The kinematics of the contour generator. Let X be a 3-D point that lies onto the smooth surface of a body part.

We derive now the constraint under which this surface point lies onto the contour generator associated to a camera. This constraint simply states that the line of sight associated with this point is tangent to the surface. Both the line of sight and the surface normal should be expressed in a common reference frame, and we choose to express these entities in the world reference frame: $(\mathbf{R}\boldsymbol{n})^\top (\mathbf{R}\boldsymbol{X} + \boldsymbol{t} - \boldsymbol{C}) = 0$, where vector $\boldsymbol{n} = \frac{\partial \boldsymbol{X}}{\partial z} \times \frac{\partial \boldsymbol{X}}{\partial \theta} = \boldsymbol{X}_z \times \boldsymbol{X}_\theta$ is normal to the surface at $\boldsymbol{X}$, and $\boldsymbol{C}$ is the camera optical center in world coordinates. The equation above becomes:

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

For any rotation, translation, and camera position, equation (3) allows to estimate $\boldsymbol{X}$ as a function of the surface parameters.

The surface of a truncated elliptical cone is parametrized by an angle θ and a height z:

$$X(\theta, z) = \begin{pmatrix} a(1+kz)\cos(\theta) \\ b(1+kz)\sin(\theta) \\ z \end{pmatrix} \tag{4}$$

where a and b are the minor and major half-axes of the elliptical cross-section, k is the tapering parameter of the cone, and $z \in [z_1, z_2]$. With this parameterization, eq. (3) can be developed to obtain a trigonometric equation of the form $F\cos\theta + G\sin\theta + H = 0$ where F, G and H depend on $\boldsymbol{\Phi}$ and C but do not depend on z. With the standard substitution $t = \tan\frac{\theta}{2}$ we obtain a second-degree polynomial:

$$(H - F)t^2 + 2Gt + (F + H) = 0 \tag{5}$$

This equation has two real solutions, t_1 and t_2, (or, equivalently, θ_1 and θ_2) whenever the camera lies outside the cone that defines the body part. Note that in the case of elliptical cones, θ_1 and θ_2 do not depend on z and the contour generator is composed of two straight lines, $X(\theta_1, z)$ and $X(\theta_2, z)$. From now on and without ambiguity, $\boldsymbol{X}$ denotes a point lying onto the contour generator.

The motion of extremal contours. The extremal contour is the projection of the contour generator. Without loss of generality, let the world frame be aligned with the camera frame. A point $\boldsymbol{x}$ of the extremal contour is therefore defined by its image coordinates: $x_1 = X_1^w/X_3^w$ and $x_2 = X_2^w/X_3^w$, with

$$\boldsymbol{X}^w = \mathbf{R}\boldsymbol{X} + \boldsymbol{t} \tag{6}$$

The velocity of x, $\boldsymbol{v}$ is computed with:

$$\boldsymbol{v} = \mathbf{J}_I\left(\dot{\mathbf{R}}\boldsymbol{X} + \dot{\boldsymbol{t}} + \mathbf{R}\dot{\boldsymbol{X}}\right) = \mathbf{J}_I(\mathbf{A} + \mathbf{B})\begin{pmatrix}\boldsymbol{\Omega}\\ \boldsymbol{V}\end{pmatrix} \tag{7}$$

where $\mathbf{A}$ and $\mathbf{B}$ are defined below and $\mathbf{J}_I$ is the classical 2×3 matrix:

$$\mathbf{J}_I = \begin{bmatrix} 1/X_3^w & 0 & -X_1^w/(X_3^w)^2 \\ 0 & 1/X_3^w & -X_2^w/(X_3^w)^2 \end{bmatrix}$$

Eq. (7) reveals that the motion of extremal contours has two components: a component due to the rigid motion of the smooth surface, and a component due to the sliding of the contour generator onto the smooth surface. The first component is:

$$\dot{\mathbf{R}}\boldsymbol{X} + \dot{\boldsymbol{t}} = \dot{\mathbf{R}}\mathbf{R}^\top(\boldsymbol{X}^w - \boldsymbol{t}) + \dot{\boldsymbol{t}} = \mathbf{A}\begin{pmatrix}\boldsymbol{\Omega}\\ \boldsymbol{V}\end{pmatrix} \tag{8}$$

where $\mathbf{A} = [-[X^w]_\times \;\; \mathbf{I}]$ and $(\boldsymbol{\Omega}, \boldsymbol{V})$ is the kinematic screw. The notation $[\boldsymbol{m}]_\times$ stands for the skew-symmetric matrix associated with a vector $\boldsymbol{m}$.

The second component can be made explicit by taking the time derivative of the contour generator constraint, i.e., eq. (3). After some algebraic manipulations, we obtain:

$$\mathbf{R}\dot{\boldsymbol{X}} = \mathbf{B}\begin{pmatrix}\boldsymbol{\Omega}\\ \boldsymbol{V}\end{pmatrix} \tag{9}$$

where $\mathbf{B} = b^{-1}\mathbf{R}\boldsymbol{X}_\theta \ (\mathbf{R}\boldsymbol{n})^\top \left[[\boldsymbol{C}-\boldsymbol{t}]_\times \quad -\mathbf{I}\right]$ is a 3×6 matrix and $b = (X^g + \mathbf{R}^T(\boldsymbol{t} - C))^T \boldsymbol{n}_\theta$ is a scalar.

The sliding of the contour generator infers an image velocity that is tangent to the extremal contour. Approaches based on the estimation of the optical flow for tracking [14] cannot take into account this tangential component of the velocity field. Within our approach this term is important and it will be argued in the experimental section below that it speeds up the convergence of the tracker by a factor of 2.

Finally we notice that the kinematic screw of a body-part can be related to the joint velocities associated with a kinematic chain [15], where $\mathbf{J}_K$ is the chain's Jacobian matrix: $(\boldsymbol{\Omega} \quad \boldsymbol{V})^\top = \mathbf{J}_K\dot{\boldsymbol{\Phi}}$. By combining this formula with eq. (7) we obtain eq. (1):

$$\boldsymbol{v} = \mathbf{J}_I(\mathbf{A}+\mathbf{B})\mathbf{J}_K\dot{\boldsymbol{\Phi}} \tag{10}$$

3 Fitting extremal contours to images

We now go back to the error function introduced in eq. (2). A well known difficulty is that one can only recover noisy and cluttered image contours and, therefore, the error function should be able to cope with this problem. One possible choice for the error funtion, that works well in practice, is the sum of the distances to the nearest image contour over all the predicted extremal contours points. Thus, the error function writes:

$$E(\mathcal{Y}, \mathcal{X}(\boldsymbol{\Phi})) = \sum_{i=1}^{N} D^2(\mathcal{Y}, \boldsymbol{x}_i(\boldsymbol{\Phi})), \tag{11}$$

where N is the number of predicted extremal contour points and D is a scalar function that returns the minimum distance to an observed contour in $\mathcal{Y}$, evaluated at image location $\boldsymbol{x}$.

The distance from a predicted extremal-contour point to the nearest image-contour point can be computed as a chamfer distance performed after edge detection. But in general one can only observe the silhouette of the actor, obtained through background subtraction, and the edges of a small number of body parts within that silhouette (figure 4). The distance we use in practice is the sum of the minimum distances to both the silhouette *and* the edges observed by all cameras. In the remainder of this section, we explain the advantages of using this particular combination of silhouettes and edges.

For clarity of the presentation, we consider the case of a single body part and we analyse the error function along an image row. Fig.3-(b) is a plot of the error function when only the silhouette is used. The chamfer distance is zero everywhere within the silhouette. Hence, the error function has a large and flat minimum – or infinitely many local minima – thus ill-suited for numerical optimization. Fig. 3-(c) is a plot of the error function when only the edges are considered. As it can be noticed, the error function is flat near the edges and

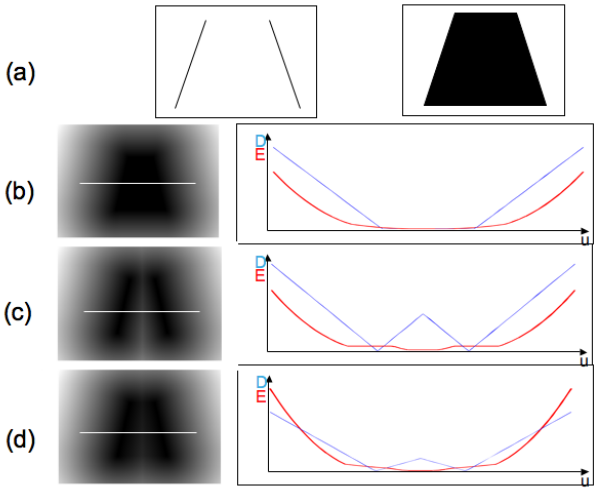

Fig. 3. (a) Observed edges (left) and silhouette (right). (b) Chamfer distance on the silhouette. (c) Chamfer distance on the edges. (d) Sum of both distances. The graphs illustrate the distance (blue or thin curve) and the error (red or bold curve) along a row (white lines).

the error function is also ill-suited. Eventually, Fig. 3-(d) is a plot of the error function when using the sum of the two previously proposed distances. The error function is never constant and there exists only one local minimum, where the model contour coincides exactly with the observed contour.

Thus, the simulteneous use of the chamfer distances of both the edges and the silhouette avoids such local minima. As explained above, minimizing the silhouette distance pushes model contours inside the image silhouettes while minimizing the edge distance attracts the model contours to high image gradients within that silhouette, without explicitly representing the contour orientations.

Now that we have chosen the error function to be minimized, we can track our model by iteratively minimizing the error in all views, using a non-linear least-squares optimization technique such as Levenberg-Marquardt. Using the results from section 2 together with a bilinear interpolation of the chamfer distance images, we compute the Jacobian analytically, which results in an efficient implementation, as described in the next section.

4 Experimental results and discussion

We performed experiments with realistic and complex human motions using a setup composed of 6 cameras that operate at 30 frames/second. The cameras are both finely synchronized (within 10^{-6}s) and operate at the same shutter speed (10^{-3}s.) thus allowing us to cope with fast motions. The 3-D human model is

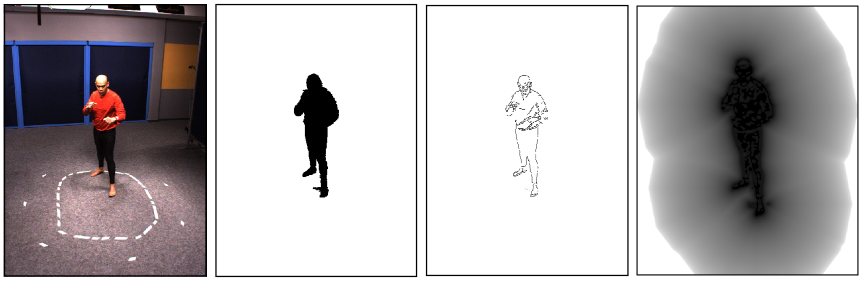

Fig. 4. From left to right: A raw image, the silhouette, the edges inside the silhouette, and the chamfer-distance image associated with the silhouette.

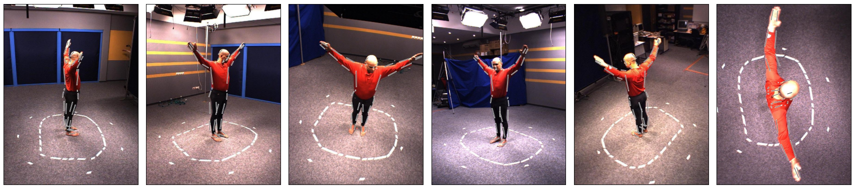

Fig. 5. A set of six calibrated cameras provides six image sequences whose frames are synchronized.

composed of 18 body parts with a total of 40 degrees of freedom[1]. We validated our tracker using realistic data sets consisting of movements performed by professionals (Fig. 1 and 6). Silhouettes and edges were extracted using standard techniques (statistical background subtraction and edge detection). In the first sequence (Figure 1) we tracked the motion over 700 frames, starting from a reference pose. In the second sequence (Figure 6), we tracked a very fast motion over 100 frames. In both cases, the optimization always converged in less than 5 iterations per frame. The RMS error on both sequences is close to one pixel. Given the roughness of the parameters modelling the person's features (length of arms, feet, thighs, etc.), this error is quite satisfactory and could probably be improved further with better estimates of the anthropometric dimensions of the human model.

We evaluated the importance of the sliding motion term in the minimization process since it was asserted to be negligible in [14]. With both synthetic and real data, we found that we could ignore the correction terms and still obtain the same results, at the expense of doubling the number of iterations, on an average. This gives experimental evidence that the correction introduced by the sliding motion of the contour generators may be important, if not critical, for real-time/best-effort implementations.

[1] 2 degrees of freedom for the head, 3 for the torso, 3 for the abdomen, 6 for the two clavicles, 6 for the two shoulders, 4 for the two elbows, 6 for the hips, and 4 at the knees, keeping the feet and the hands rigidly attached to the ankles and forearms.

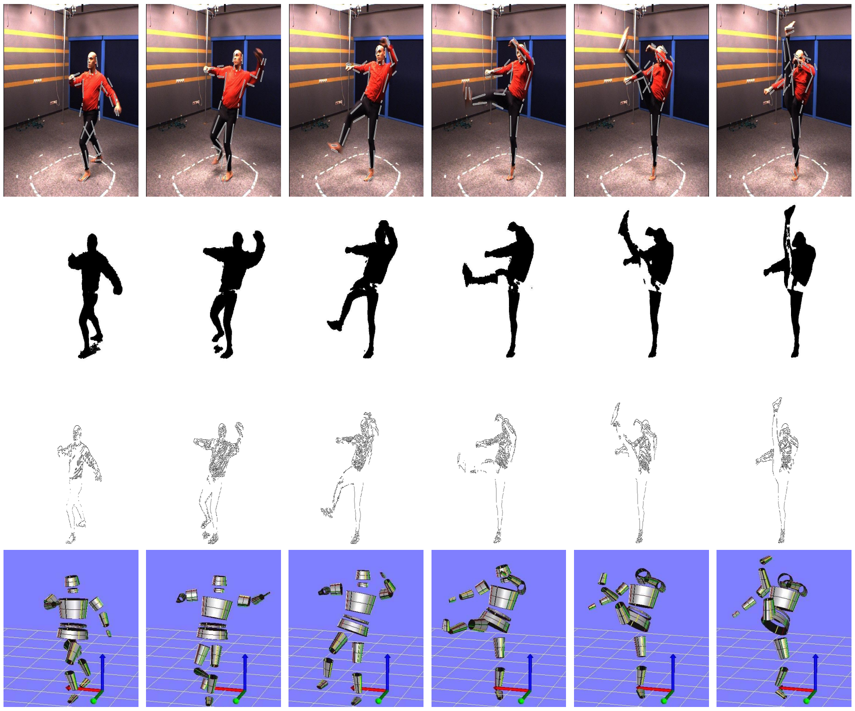

Fig. 6. Tracking a "taekwondo" sequence. From top to bottom: Extremal contours predicted from the previously estimated pose; Silhouettes extracted with a background subtraction algorithm; Edges inside the silhouettes, and the estimated pose of the articulated model.

With our current algorithms we did not restrict the joint angles to biomechanically feasible limits. As a result, most of our tracker failures occurred because of incorrect assignments during matching, which resulted in collisions between body parts. We believe we can solve this problem by implementing collision detection and collision prevention more carefully. Another important issue that should be addressed in future work, is the automatic calibration of the parameters of our human-body model. Obtaining optimal values for all the constant geometric and kinematic parameters in the anthropomorphic model will be important for evaluating and improving further the quality, robustness, and precision of our tracker.

5 Summary and Conclusion

We described a method for using image silhouettes and edges from several cameres in order to estimate the articulated motion of a person. Our approach works well with relatively difficult motions, using non-textured clothes with shadows and folds. We presented a derivation of the image Jacobian for that case, and demonstrated experimentally that the resulting tracker converges in fewer (typically less than five) iterations per frame, compared to the classical rigid-motion approximation.

Future work will be devoted to extend the method to other body part shapes such as the head, hands and feet, to combine information form the contours with point features and textures, when they are available, to fit the constant geometric and kinematic parameters of our models automatically, and to feed the results into a Kalman or particle-filter representation of human dynamics.

References

1. Gavrila, D.M.: The visual analysis of human movement: A survey. Computer Vision and Image Understanding **73** (1999) 82–98
2. Bregler, C., Malik, J., Pullen, K.: Twist based acquisition and tracking of animal and human kinematics. International Journal of Computer Vision **56** (2004) 179–194
3. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: Computer Vision and Pattern Recognition. (2000) 2126–2133
4. Hilton, A.: Towards model-based capture of a persons shape, appearance and motion. In: Proceedings of the IEEE International Workshop on Modelling People. (1999)
5. Yan, J., Pollefeys, M.: A factorization approach to articulated motion recovery. In: Conference on Computer Vision and Pattern Recognition. Volume 2. (2005) 815–821
6. Drummond, T., Cipolla, R.: Real-time tracking of highly articulated structures in the presence of noisy measurements. In: ICCV. (2001) 315–320
7. Sminchisescu, C., Telea, A.: Human pose estimation from silhouettes. a consistent approach using distance level sets. In: WSCG International Conference on Computer Graphics, Visualization and Computer Vision. (2002)
8. Delamarre, Q., Faugeras, O.: 3d articulated models and multi-view tracking with physical forces. Computer Vision and Image Understanding **81** (2001) 328–357
9. Niskanen, M., Boyer, E., Horaud, R.: Articulated motion capture from 3-d points and normals. In Clocksin, Fitzgibbon, T., ed.: British Machine Vision Conference. Volume 1., Oxford, UK, BMVA, British Machine Vision Association (2005) 439–448
10. Blake, A., Isard, M.: Active Contours. Springer-Verlag (1998)
11. Agarwal, A., Triggs, B.: Learning to track 3d human motion from silhouettes. In: International Conference on Machine Learning, Banff (2004) 9–16
12. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. IEEE Trans. Pattern Analalysis Machine Intelligence **24** (2002) 932–946
13. Martin, F., Horaud, R.: Multiple camera tracking of rigid objects. International Journal of Robotics Research **21** (2002) 97–113
14. Rosten, E., Drummond, T.: Rapid rendering of apparent contours of implicit surfaces for real-time tracking. In: British Machine Vision Conference. Volume 2. (2003) 719–728
15. McCarthy, J.M.: Introduction to Theoretical Kinematics. MIT Press, Cambridge (1990)