Silhouette Based Human Motion Estimation

Bodo Rosenhahn[1], Reinhard Klette[1] and Gerald Sommer[2]

[1] University of Auckland (CITR)
Computer Science Department
Private Bag 92019 Auckland, New Zealand
bros028@cs.auckland.ac.nz, r.klette@cs.auckland.ac.nz
[2] Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40, 24098 Kiel, Germany
gs@ks.informatik.uni-kiel.de

Abstract. In this contribution we present an algorithm for 2D-3D pose estimation of human beings. A human torso is modeled in terms of free-form surface patches which are extended with joints inside the surface. We determine the 3D pose and the angles of the arm joints from image silhouettes of the torso. This silhouette based approach towards human motion estimation is illustrated by experimental results for monocular or stereo image sequences.

1 Introduction

Modeling and tracking of human motion from video sequences is an increasingly important field of research with applications in sports sciences, medicine, animation (avatars) or surveillance. E. Muybridge is known as the pioneer in human motion capturing with his famous experiments in 1887 called *Animal Locomotion*. In recent years, many techniques for human motion tracking have been proposed which are fairly effective [2, 5], but they often use simplified models of the human body by applying ellipsoidal, cylindrical or skeleton models and do not use a realistic surface model. The reader is referred to [4] for a recent survey on marker-less human motion tracking.

In this work we present and discuss a human motion capturing system which estimates the pose and angle configuration of a human body captured in image sequences. Contrary to other works we apply a 2-parametric surface representation [3], allow full perspective camera models, and use the extracted silhouette of the body as the only image information. Our algorithms are fast (400ms per frame), and we present experiments on monocular and stereo image sequences. The scenario is visualized in the left of figure 1. As it can be seen, we use a model of the human torso with its arms, and model it by using two free-form surface patches. The first patch (modeling the torso) contains 57×21 nodes and the second (modeling the arms) contains 81×21 nodes. Each arm contains 4 joints, so that we have to deal with 8 joint angles and 6 unknowns for the rigid motion resulting in 14 unknowns. The right of figure 1 gives the names of the used joints for the diagrams in the experiments.

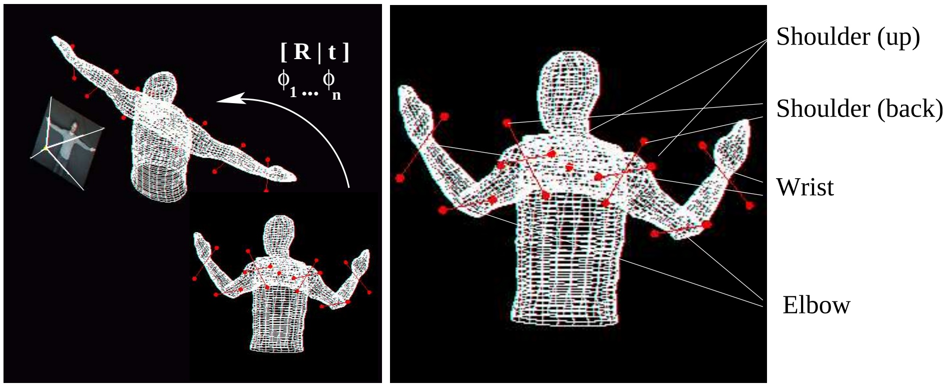

Fig. 1. Left: The pose scenario: the aim is to estimate the pose $\boldsymbol{R}$, $\boldsymbol{t}$ and the joint angles ϕ_i. Right: The names of the used joints.

This contribution continues work reported in [7, 8] on point-based, contour- and surface-based pose estimation. But whereas in these works only rigid objects are discussed or only a point-based representation scheme for modeling kinematic chains is used, in this work we want to overcome these previous limitations by introducing an approach for pose estimation of free-form surfaces, coupled with kinematic chains. It is applied to marker-less human motion tracking. We start with recalling foundations and introduce twists which are used to model rigid motions and joints. Then we continue with free-form contours and surfaces, and define a basic approach for pose estimation, followed by extensions and experiments. We conclude with a brief discussion.

2 Foundations

Clifford or geometric algebras [9] can be used to deal with geometric aspects of the pose problem. We only list a few properties which are important for our studies. The elements in geometric algebras are called multivectors which can be multiplied by using a geometric product. It allows a coordinate-free and dense symbolic representation. For modeling the pose problem, we use the conformal geometric algebra (CGA). The CGA is build up on a conformal model which is coupled with a homogeneous model to deal with kinematics and projective geometry simultaneously. In conclusion, we deal with the Euclidean, kinematic and projective space in a uniform framework and can therefore cope with the pose problem in an efficient manner. In the equations we will use the inner product, $\cdot$, the outer product, $\wedge$, the commutator, $\underline{\times}$, and anticommutator, $\overline{\times}$, product, which can be derived from the geometric product. Though we will also present equations formulated in conformal geometric algebra, we only explain these symbolically and want to refer to [7] for more detailed information.

2.1 Point based pose estimation

For 2D-3D point based pose estimation we use constraint equations which compare 2D image points with 3D object points. Assume an image point $\boldsymbol{x}$ and the optical center $\boldsymbol{O}$. These define a 3D projection ray, $\underline{\boldsymbol{L}}_x = \mathbf{e} \wedge (\boldsymbol{O} \wedge \boldsymbol{x})$, as Plücker line [6]. The motor $\boldsymbol{M}$ is defined as exponential of a twist Ψ, $\boldsymbol{M} = \exp(-\frac{\theta}{2}\Psi)$, and formalizes the unknown rigid motion as a screw motion [6]. The motor $\boldsymbol{M}$ is applied on an object point $\underline{\boldsymbol{X}}$ as versor product, $\underline{\boldsymbol{X}}' = \boldsymbol{M}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{M}}$, where $\widetilde{\boldsymbol{M}}$ represents the so-called reverse of $\boldsymbol{M}$. Then the rigidly transformed object point, $\underline{\boldsymbol{X}}'$, is compared with the reconstructed line, $\underline{\boldsymbol{L}}_x$, by minimizing the error vector between the point and the line. This specifies a constraint equation in geometric algebra:

$$(\boldsymbol{M}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{M}}) \underline{\times} (\mathbf{e} \wedge (\boldsymbol{O} \wedge \boldsymbol{x})) = 0.$$

Note, that we deal with a 3D formalization of the pose problem. The constraint equations can be solved by linearization (i.e. solving the equations for the twist-parameters which generate the screw motion) and by applying the Rodrigues formula for a reconstruction of the group action [6]. Iteration leads to a gradient descent method in 3D space. This is presented in [7] in more detail, where similar equations have been introduced to compare 3D points with 2D lines (3D planes) and 3D lines with 2D lines (3D planes). Pose estimation can be performed in real-time and we need 2ms on a Linux 2GHz machine to estimate a pose based on 100 point correspondences.

Joints along the kinematic chain can be modeled as special screws with no pitch. In [7] we have shown, that the twist then corresponds to a scaled Plücker line, $\Psi = \theta\underline{\boldsymbol{L}}$ in 3D space, which gives the location of the general rotation. Because of this relation it is simple to move joints in space and they can be transformed by a motor $\boldsymbol{M}$ in a similar way such as plain points, $\Psi' = \boldsymbol{M}\Psi\widetilde{\boldsymbol{M}}$.

2.2 Contour-based pose estimation

We now model free-form contours and their embedding into the pose problem. As it turned out, Fourier descriptors are very useful, since they are a special case of so-called *twist-generated* curves which we used to model cycloidal curves (cardioids, nephroids and so forth) within the pose problem [7]. The later introduced pose estimation algorithm for surface models goes back onto a contour based method. Therefore, a brief recapitulation of our former works on contour based pose estimation is of importance. The main idea is to interpret a 1-parametric 3D closed curve as three separate 1D signals which represent the projections of the curve along the x, y and z axis, respectively. Since the curve is assumed to be closed, the signals are periodic and can be analyzed by applying a 1D discrete Fourier transform (1D-DFT). The inverse discrete Fourier transform (1D-IDFT) enables us to reconstruct low-pass approximations of each signal. Subject to the sampling theorem, this leads to the representation of the 1-parametric 3D curve $C(\phi)$ as

$$C(\phi) = \sum_{m=1}^{3} \sum_{k=-N}^{N} \boldsymbol{p}_k^m \exp\left(\frac{2\pi k\phi}{2N+1}\boldsymbol{l}_m\right).$$

The parameter m represents each dimension and the vectors $\boldsymbol{p}_k^m$ are phase vectors obtained from the 1D-DFT acting on dimension m. In this equation we have replaced the imaginary unit $i = \sqrt{-1}$ by three different rotation planes, represented by the bivectors $\boldsymbol{l_i}$, with $\boldsymbol{l_i}^2 = -1$. Using only a low-index subset of the Fourier coefficients results in a low-pass approximation of the object model which can be used to regularize the pose estimation algorithm. For pose estimation this model is then combined with a version of an ICP-algorithm [10].

2.3 Silhouette-based pose estimation of free-form surfaces

To model surfaces, we assume a two-parametric surface [3] of the form

$$F(\phi_1, \phi_2) = \sum_{i=1}^{3} f^i(\phi_1, \phi_2)\mathbf{e}_i,$$

with three 2D functions $f^i(\phi_1, \phi_2) : \mathbb{R}^2 \to \mathbb{R}$ acting on the different Euclidean base vectors $\mathbf{e}_i$ $(i = 1, \ldots, 3)$. The idea behind a two-parametric surface is to assume two independent parameters ϕ_1 and ϕ_2 to sample a 2D surface in 3D space. For a discrete number of sampled points, $f^i_{n_1,n_2}$, $(n_1 \in [-N_1, N_1]; n_2 \in [-N_2, N_2];\ N_1, N_2 \in \mathbb{N},\ i = 1, \ldots, 3)$ on the surface, we can now interpolate the surface by using a 2D discrete Fourier transform (2D-DFT) and then apply an inverse 2D discrete Fourier transform (2D-IDFT) for each base vector separately. Subject to the sampling theorem, the surface can be written as a Fourier representation,

$$F(\phi_1, \phi_2) = \sum_{i=1}^{3} \sum_{k_1=-N_1}^{N_1} \sum_{k_2=-N_2}^{N_2} \boldsymbol{p}^i_{k_1,k_2} \exp\left(\frac{2\pi k_1 \phi_1}{2N_1 + 1}\boldsymbol{l_i}\right) \exp\left(\frac{2\pi k_2 \phi_2}{2N_2 + 1}\boldsymbol{l_i}\right).$$

The complex Fourier coefficients are contained in the vectors $\boldsymbol{p}^i_{k_1,k_2}$ that lie in the plane spanned by $\boldsymbol{l_i}$. We will again call them phase vectors. These vectors can be obtained by a 2D-DFT of the sample points $f^i_{n_1,n_2}$ on the surface. We now continue with the algorithm for silhouette-based pose estimation of surface models.

```
Surface based pose estimation
  Reconstruct projection rays from image points
  Project the low-pass object model in the virtual image
  Estimate the 3D silhouette
  Apply contour based pose estimation algorithm
    ICP
      Estimate the nearest point on the 3D contour to each ray
      Use correspondence set to estimate the contour pose
      Transform the contour model
  Transform the surface model
  Increase low-pass approximation of the surface model
```

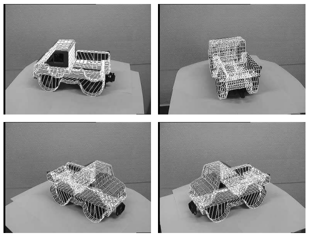

Fig. 2. Left: The algorithm for pose estimation of surface models. Right: A few example images of a tracked car model on a turn-table.

We assume a properly extracted silhouette (i.e., in a frame of the sequence) of our object (i.e., the human body). To compare points on the image silhouette we consider rim points on the surface model (i.e., which are on an occluding boundary of the object). This means we work with the 3D silhouette of the surface model with respect to the camera. To obtain this, we project the 3D surface on a virtual image. Then the contour is calculated and from the image contour the 3D silhouette of the surface model is reconstructed. The contour model is then applied within the contour-based pose estimation algorithm. Since aspects of the surface model are changing during ICP-cycles, a new silhouette will be estimated after each cycle to deal with occlusions within the surface model. The algorithm for pose estimation of surface models is summarized in figure 2 and it is discussed in [8] in more detail.

3 Human Motion Estimation

We now introduce how to couple kinematic chains within the surface model and present a pose estimation algorithm which estimates the pose and angle configurations simultaneously.

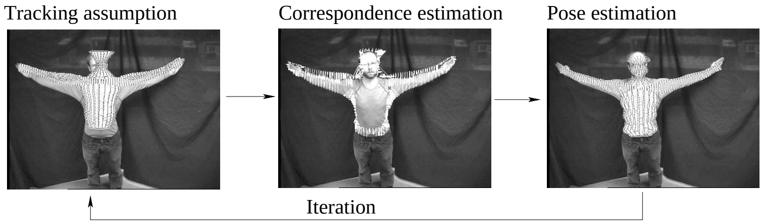

Fig. 3. The basic algorithm: Iterative correspondence and pose estimation.

A surface is given in terms of three 2-parametric functions with respect to the parameters ϕ_1 and ϕ_2. Furthermore, we assume a set of joints J_i. By using an extra function $\mathcal{J}(\phi_1, \phi_2) \to [J_i | J_i : i\text{th. joint}]$, we are able to give every node a joint list along the kinematic chain. Note, that we use $[,]$ and not $\{,\}$, since the joints are given ordered along the kinematic chain. Since the arms contain two kinematic chains (for the left and right arm separately), we introduce a further index to separate the joints on the left arm from the ones on the right arm. The joints themselves are represented as objects in an extra field (a look-up table) and their parameters can be accessed immediately from the joint index numbers. Furthermore, it is possible to transform the location of the joints in space (as clarified in section 2). For pose estimation of a point $\underline{\boldsymbol{X}}_{n,i_n}$ attached to the nth joint, we generate constraint equations of the form

$$(\boldsymbol{M}(\boldsymbol{M}_1 \ldots \boldsymbol{M}_n \underline{\boldsymbol{X}}_{n,i_n} \widetilde{\boldsymbol{M}_n} \ldots \widetilde{\boldsymbol{M}_1})\widetilde{\boldsymbol{M}}) \underline{\times} \mathbf{e} \wedge (\boldsymbol{O} \wedge \boldsymbol{x}_{n,i_n}) = 0.$$

To solve a set of such constraint equations we linearize the motor $\boldsymbol{M}$ with respect to the unknown twist Ψ and the motors $\boldsymbol{M}_i$ with respect to the unknown angles θ_i. The twists Ψ_i are known a priori.

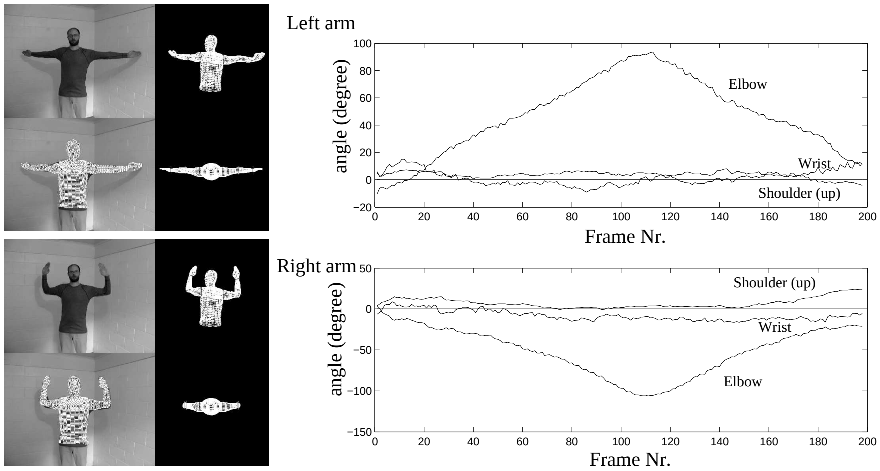

Fig. 4. Left: First pose results with a 6 DOF kinematic chain. Right: Angles of the left and right arm during the tracked image sequence.

The basic pose estimation algorithm is visualized in figure 3: We start with simple image processing steps to gain the silhouette information of the person by using a color threshold and a Laplace operator. Then we project the surface mesh in a virtual image and estimate its 3D contour. Each point on the 3D contour carries a given joint index. Then we estimate the correspondences by using an ICP-algorithm, generate the system of equations, solve them, transform the object and its joints and iterate this procedure. During iteration we start with a low-pass object representation and refine it by using higher frequencies. This helps to avoid local minima during iteration.

First results of the algorithm are shown on the left of figure 4: The figure contains two pose results; it shows on each quadrant the original image and overlaid the projected 3D pose. The other two images show the estimated joint angles in a virtual environment to visualize the error between the ground truth and the estimated pose. The tracked image sequence contains 200 images. In this sequence we use just three joints on each arm and neglect the shoulder (back) joint. The right diagram of figure 4 shows the estimated angles of the joints during the image sequence. The angles can easily be identified with the sequence. Since the movement of the body is continuous, the estimated curves are also relatively smooth.

Then we extend the model to a 8DOF kinematic chain and add a joint on the shoulder which allows the arms to move backwards and forwards. Results of the same sequence are shown in figure 5. As it can be seen, the observation of the pose overlaid with the image data appear to be good, but in a simulation environment it can be seen, that estimated joints are quite noisy. The reason for the depth sensitivity lies in the used image information: Figure 6 shows two images of a human with different arm positions. It can be seen, that

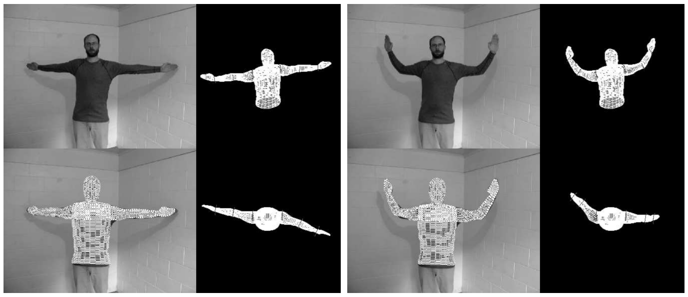

Fig. 5. First pose results with a 8 DOF kinematic chain.

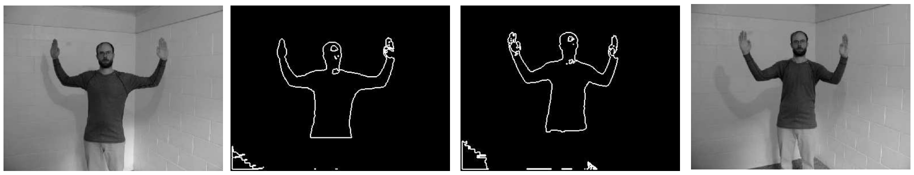

Fig. 6. The silhouette for different arm poses of the kinematic chain.

the estimated silhouettes look quite similar. This means, that the used image features are under-determined in their interpretation as 3D pose configuration. This problem can not be solved in an algorithmic way and is of geometric nature. To overcome this problem we decided to continue with a stereo setup. The basic idea is, that the geometric non-uniquenesses can be avoided by using several cameras observing the scene from different perspectives. Since we reconstruct rays from image points, we have to calibrate the cameras with respect to one fixed world coordinate system. Then it is unimportant for which camera a ray is reconstructed and we are able to combine the equations from both cameras into one system of equations and estimate the pose and arm angles simultaneously.

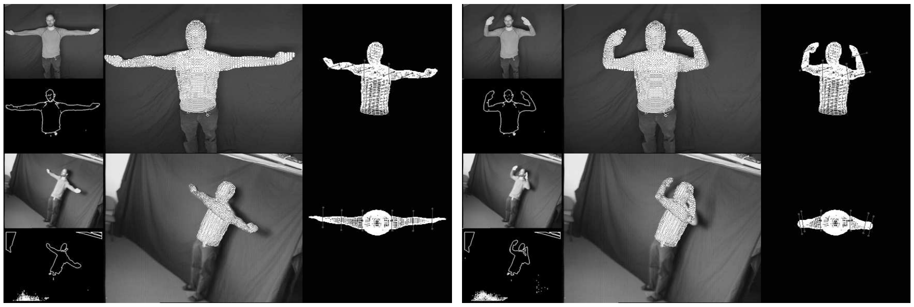

Fig. 7. Example images of a first stereo experiment.

Figure 7 shows example images of our stereo implementation. In each segment, the left images show the original and filtered image of each camera. The middle images show pose results in both cameras and the right images show the pose results in a virtual environment. It can be seen that the results improved.

4 Discussion

This contribution presents an approach for silhouette-based pose estimation of free-form surfaces coupled with kinematic chains. We use our previous work, dealing with 2D-3D pose estimation of points, free-form contours and free-form surfaces and describe how to extend the approach to kinematic chains. In the experiments it turns out that pure silhouette information is not sufficient for accurate pose estimation since the extracted silhouette and its interpretation as 3D pose is under-determined. Therefore, we move on to a multi-view scenario and illustrate that the pose results can be improved remarkably in a stereo setup. Experiments have been done with image sequences between 100 and 500 frames. Further work will continue with an extension of the multi-camera set-up.

Acknowledgments
This work has been supported by the EC Grant IST-2001-3422 (VISATEC) and by the DFG grant RO 2497/1-1.

References

1. Arbter K. and Burkhardt H. Ein Fourier-Verfahren zur Bestimmung von Merkmalen und Schätzung der Lageparameter ebener Raumkurven. *Informationstechnik*, Vol. 33, No. 1, pp. 19-26, 1991.
2. Bregler C. and Malik J. Tracking people with twists and exponential maps. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, pp. 8-15, 1998.
3. Campbell R.J. and Flynn P.J. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding (CVIU)*, Vol. 81, pp. 166-210, 2001.
4. Gavrilla D.M. The visual analysis of human movement: A survey *Computer Vision and Image Understanding*, Vol. 73 No. 1, pp. 82-92, 1999.
5. Mikic I., Trivedi M, Hunter E, and Cosman P. Human body model acquisition and tracking using voxel data *International Journal of Computer Vision (IJCV)*, Vol. 53, Nr. 3, pp. 199–223, 2003.
6. Murray R.M., Li Z. and Sastry S.S. A Mathematical Introduction to Robotic Manipulation. *CRC Press*, 1994.
7. Rosenhahn B. Pose Estimation Revisited. (PhD-Thesis) *Technical Report 0308, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik*, 2003. Available at `www.ks.informatik.uni-kiel.de`
8. Rosenhahn B., Perwass C. and Sommer G. Pose estimation of free-form surface models. In *Pattern Recognition, 25th DAGM Symposium*, B. Michaelis and G. Krell (Eds.), Springer-Verlag, Berling, LNCS 2781, pp. 574-581.
9. Sommer G., editor. Geometric Computing with Clifford Algebra. *Springer Verlag*, Berlin, 2001.
10. Zang Z. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, Vol. 13, No. 2, pp. 119-152, 1999.