Pose Estimation of 3D Free-form Contours in Conformal Geometry

Bodo Rosenhahn, Christian Perwass and Gerald Sommer
Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Preußerstrasse 1-9, 24105 Kiel
Germany
{bro,chp,gs}@ks.informatik.uni-kiel.de

Abstract

In this article we discuss the 2D-3D pose estimation problem of 3D free-form contours. In our scenario we observe objects of any 3D shape in an image of a calibrated camera. Pose estimation means to estimate the relative position and orientation (containing a rotation $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$) of the 3D object to the reference camera system. The fusion of modeling free-form contours within the pose estimation problem is achieved by using the conformal geometric algebra. Free-form contours are modeled as 3D Fourier descriptors and combined with an ICP (Iterative Closest Point) algorithm they are embedded in the pose problem as unique entities. In this work we further concentrate on modeling multiple object contours (coming along with object occlusions, etc.) and the modeling of object deformations. Object deformations are achieved by combining kinematic chains within Fourier descriptors.

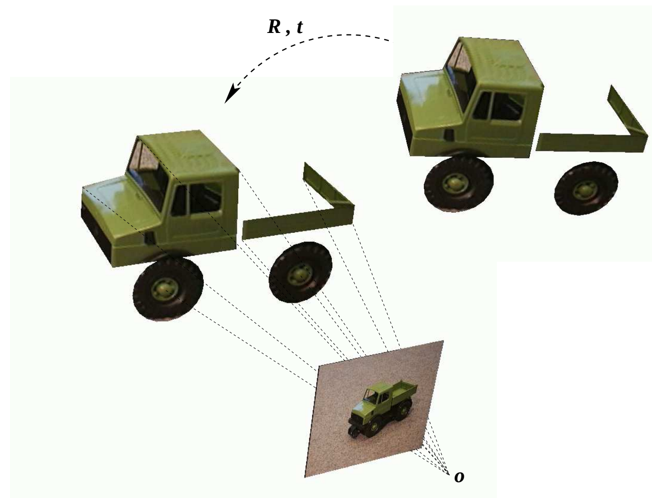

Figure 1: The scenario. The assumptions are the projective camera model, the model of the object and corresponding extracted entities on the image plane. The aim is to find the pose ($\boldsymbol{R}$, $\boldsymbol{t}$) of the model, which leads to the best fit of the object with the actually extracted entities.

1 Introduction and preliminary work

Pose estimation itself is one of the oldest computer vision problems. Algebraic solutions with different camera models have been proposed for several variations of this problem. Pioneering work was done in the 80's and 90's by Lowe [Lowe 1980; Lowe 1987], Grimson [Grimson 1990] and others. In their work, point correspondences are used. More abstract entities can be found in [Zerroug et al 1996; Kriegmann et al 1992; Bregler et al 1998]. Discussed entities are circles, cylinders, kinematic chains or other multi-part curved objects. Works concerning free-form curves can be found in [Drummond et al 2000; Stark 1996]. In their work, contour point sets, affine snakes, or active contours are used for visual servoing.

Our recent work [Rosenhahn et al 2002] can be summarized in the scenario of figure 1: We assume object features like 3D points, 3D lines, 3D spheres, 3D circles or kinematic chain segments of a reference model. Further, we assume extracted corresponding features in an image of a calibrated camera. The aim is to find the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ of the object, which leads to the best fit of the reference model with the actually extracted entities. To relate 2D image information to 3D entities we interpret an extracted image entity, resulting from the perspective projection, as a one dimension higher entity, gained by projective reconstruction from the image entity. This idea will be used to formulate the scenario as a pure kinematic problem.

The problem with feature based pose estimation is that there exist many scenarios (e.g. in natural environments) in which it is not possible to extract point-like features such as corners or edges. Then there is need to deal for example with the silhouette of the object as a whole, instead of sparse local features on the silhouette. Besides, there exist 3D objects which cannot adequately be represented by primitive object features such as points, lines or circles. These are the scenarios we address in this paper through the use of free-form contours.

2 The pose problem in conformal geometric algebra

This section concerns the formalization of the pose problem in conformal geometric algebra. Geometric algebras are the language we use for our pose problem and the main arguments for using this language are its dense symbolic representation and its coupling of projective and kinematic geometry. We will first introduce the basic notation of conformal geometric algebra and then continue with the modeling of entities and their kinematic transformations. We make use of it to model free-form contours in the 3D space. Then we combine this formalization within the pose estimation problem.

Introduction to conformal geometric algebra
We now introduce the main properties of the conformal geometric algebra (CGA) [Li et al 2000]. The aim is to clarify

the notations. A more detailed introduction concerning geometric algebras can be found in [Sommer 2001].

The main idea of geometric algebras $\mathcal{G}$ is to define a product on basis vectors, which extends the linear vector space V of dimension n to a linear space of dimension 2^n. The elements are so-called multivectors as higher order algebraic entities in comparison to vectors of a vector space as first order entities. A geometric algebra is denoted as $\mathcal{G}_{p,q}$ with $n = p + q$. Here p and q indicate the numbers of basis vectors which square to $+1$ and -1, respectively. The product defining a geometric algebra is called *geometric product* and is denoted by juxtaposition, e.g. $\boldsymbol{uv}$ for two multivectors $\boldsymbol{u}$ and $\boldsymbol{v}$. Operations between multivectors can be expressed by special products, called *inner* $\cdot$, *outer* $\wedge$, *commutator* $\underline{\times}$ and *anticommutator* $\overline{\times}$ product.

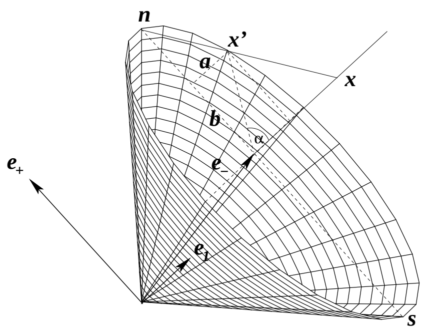

Figure 2: Visualization of the homogeneous model for stereographic projections for the 1D case. All stereographic projected points are on a cone, which is a null-cone in the Minkowski space.

The idea behind conformal geometry is to interpret points as *stereographic projected* points. The rule for a stereographic projection has a nice geometric description and is visualized for a homogeneous 1D case in figure 2: Think of the earth as a transparent sphere, intersected on the equator by an *equatorial plane.* Now imagine a light bulb at the *north pole* $\boldsymbol{n}$, which shines through the sphere. Each point on the sphere casts a shadow on the paper, and that is where it is drawn on the map. The basic formulas for projecting points in space on the sphere and vice versa are for example given in [Needham 1997]. Using a homogeneous model for stereographic projected points leads to a cone in the space, which is visualized in figure 2. This cone is spanned by one positive and one negative squaring basis vector, introducing a Minkowski metric and leading to a null-cone.

The conformal geometric algebra $\mathcal{G}_{4,1}$ (CGA) [Li et al 2000] is suited to describe conformal geometry and it contains spheres as entities and a rich set of geometric manipulations. The point at infinity, $\boldsymbol{n} \simeq \mathbf{e}$, and the origin, $\boldsymbol{s} \simeq \mathbf{e}_0$, are special elements and define a null space in the conformal geometric algebra. Evaluating the point $\underline{\boldsymbol{x}}$ on the cone leads to

$$\underline{\boldsymbol{x}} \quad = \quad \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathbf{e} + \mathbf{e}_0.$$

This point representation can be interpreted as a sphere with radius zero. A general sphere, defined by the center $\boldsymbol{p}$ and the radius ρ, is given as

$$\underline{\boldsymbol{s}} \quad = \quad \boldsymbol{p} + \frac{1}{2}(\boldsymbol{p}^2 - \rho^2)\mathbf{e} + \mathbf{e}_0,$$

and a point $\underline{\boldsymbol{x}}$ is on a sphere $\underline{\boldsymbol{s}}$ iff $\underline{\boldsymbol{x}} \cdot \underline{\boldsymbol{s}} = 0$. The multivector concepts of geometric algebras then allow to intersect spheres and define elements, like points, lines, planes or circles as entities, generated from spheres.

Rotations are represented by rotors, $\boldsymbol{R} = \exp\left(-\frac{\theta}{2}\boldsymbol{l}\right)$. The components of the rotor $\boldsymbol{R}$ are the unit bivector $\boldsymbol{l}$ which represents the dual of the rotation axis, and the angle θ which represents the amount of the rotation. The rotation of an entity can be performed by its spinor product $\underline{\boldsymbol{X}}' = \boldsymbol{R}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{R}}$. The multivector $\widetilde{\boldsymbol{R}}$ denotes the reverse of $\boldsymbol{R}$. A translation can be expressed by a translator, $\boldsymbol{T} = (1 + \frac{\mathbf{e}\boldsymbol{t}}{2}) = \exp\left(\frac{\mathbf{e}\boldsymbol{t}}{2}\right)$. To estimate the rigid body motion (containing a rotor $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$), we follow e.g. [Murray et al 1994]: A rigid body motion can be expressed by a rotation about a line in space. This results from the fact that for every $g \in SE(3)$ exists a $\xi \in se(3)$ and a $\theta \in \mathbb{R}$ such that $g = \exp(\xi\theta)$. The element ξ is also called a *twist.* The motor $\boldsymbol{M}$ describing a twist transformation has the general form $\boldsymbol{M} = \boldsymbol{TR}\widetilde{\boldsymbol{T}}$, denoting the inverse translation, rotation and back translation, respectively. But whereas in Euclidean geometry, Lie algebras and Lie groups are only applied on point concepts, the motors and twists of the CGA can also be applied on other entities, like lines, planes, circles, spheres, etc.

Constraint equations for pose estimation

Now we express the 2D-3D pose estimation problem: *a transformed object entity has to lie on a spatial entity, projectively reconstructed from an image entity.* Let $\underline{\boldsymbol{X}}$ be an object point and $\underline{\boldsymbol{L}}$ be an object line, given in CGA. The (unknown) transformations of the entities can be described as $\boldsymbol{M}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{M}}$ and $\boldsymbol{M}\underline{\boldsymbol{L}}\widetilde{\boldsymbol{M}}$, respectively. Let $\boldsymbol{x}$ be an image point and $\boldsymbol{l}$ be an image line on a projective plane. The projective reconstruction of an image point in CGA can be written as $\underline{\boldsymbol{L}}_x = \mathbf{e} \wedge \boldsymbol{O} \wedge \boldsymbol{x}$. The entity $\underline{\boldsymbol{L}}_x$ is a circle, containing the vector $\boldsymbol{O}$ as the optical center of the camera, see e.g. figure 1, the image point $\boldsymbol{x}$ and the vector $\mathbf{e}$ as the point at infinity. This leads to a reconstructed projection ray. Similarly leads $\underline{\boldsymbol{P}}_l = \mathbf{e} \wedge \boldsymbol{O} \wedge \boldsymbol{l}$ to a reconstructed projection plane in CGA. Collinearity and coplanarity can be described by the commutator and anticommutator products. Thus, the constraint equations of pose estimation from image points read

$$\underbrace{\underbrace{(\boldsymbol{M}\underbrace{\underline{\boldsymbol{X}}}_{\text{object point}}\widetilde{\boldsymbol{M}})}_{\text{rigid motion of the object point}} \;\underline{\times}\; \underbrace{\mathbf{e}\wedge(\boldsymbol{O}\wedge\boldsymbol{x})}_{\text{projection ray, reconstructed from the image point}}}_{\text{collinearity of the transformed object point with the reconstructed line}} = 0.$$

Constraint equations which relate 2D image lines to 3D object points, or 2D image lines to 3D object lines, can be expressed in a similar manner. Note: The constraint equations in the unknown motor $\boldsymbol{M}$ express a distance measure which has to be zero. This is e.g. shown in [Rosenhahn et al 2000].

Fourier descriptors in CGA

Fourier descriptors are often used for object recognition [Granlund 1972; Zahn et al 1996] and affine pose estimation [Arbter et al 1991; Reiss 1993] of closed contours. We now concern the formalization of Fourier descriptors in CGA to combine it with our previous introduced pose estimation constraints. Let

$$\boldsymbol{R}_i^{\phi} \quad := \quad \exp(-\frac{\pi u_i \phi}{T}\boldsymbol{l}),$$

where $T \in \mathbb{R}$ is the length of the closed curve, $u_i \in \mathbb{Z}$ is a frequency number and $\boldsymbol{l}$ is a unit bivector which defines the rotation plane. Furthermore is $\widetilde{\boldsymbol{R}}_i^{\phi} = \exp(\pi u_i \phi / T\,\boldsymbol{l})$. Recall that $\boldsymbol{l}^2 = -1$ and we can therefore write the exponential function as $\exp(\phi \boldsymbol{l}) = \cos(\phi) + \sin(\phi)\boldsymbol{l}$. We can now write any closed, planar curve $C(\phi)$ as a series expansion

$$C(\phi) = \lim_{N\to\infty} \sum_{k=-N}^{N} \boldsymbol{p}_k \exp\left(\frac{2\pi k \phi}{T} \boldsymbol{l}\right) = \lim_{N\to\infty} \sum_{k=-N}^{N} \boldsymbol{R}_k^{\phi} \, \boldsymbol{p}_k \, \widetilde{\boldsymbol{R}}_k^{\phi}.$$

This can be interpreted as a Fourier series expansion, where we have replaced the imaginary unit $i = \sqrt{-1}$ with $\boldsymbol{l}$ and the complex Fourier series coefficients with vectors that lie in the plane spanned by $\boldsymbol{l}$. The vectors $\boldsymbol{p}_k$ are the phase vectors. In general it may be shown that for every closed curve there is a unique set of phase vectors $\{\boldsymbol{p}_k\}$ that parameterizes the curve. In figure 3 a closed curve is shown in the yz-

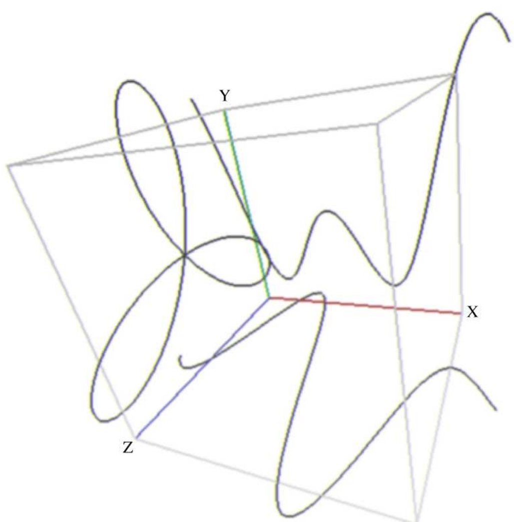

Figure 3: Projections of a curve interpreted as time dependend function.

plane. Suppose that instead of $C(\phi)$ we consider $C_S(\phi) := C(\phi) + 2\pi\phi/T\, \boldsymbol{e}_1$, where $\boldsymbol{e}_1$ is the unit vector along the x-axis. The x-axis can be interpreted as an additional time axis. If we project $C_S(\phi)$ onto the xy-plane and xz-plane, we obtain the two other curves shown. This visualizes the well known fact that we can regard any periodic function in a space of dimension n as the projection of a closed curve in a space of dimension $n+1$.

The phase vectors $\{\boldsymbol{p}_k\}$ are also called *Fourier descriptors*. It has long been known that one can also construct affine invariant Fourier descriptors [Granlund 1972], that is, entities that describe a closed curve and stay invariant under affine transformations of the curve. We also attempted to perform a projective pose estimation via Fourier descriptors. Unfortunately, there are two major problems. First of all, if a closed curve is projected projectively, then the projected curve will not be sampled in the same way as the original curve. This already distorts the Fourier descriptors. Secondly, going through the equations we found that in order to solve the projective pose estimation problem via Fourier descriptors, one has to find analytic solutions to n^{th} degree polynomials. Since this is not possible in general, we cannot follow this approach. We therefore investigated a different approach for the pose estimation of projected closed curves, which leads to a kinematic description of the pose problem.

3 Pose estimation of free-form contours

We assume a given closed, discretizied 3D curve, that is a 3D contour C with $2N$ sampled points in both the spatial and spectral domain with phase vectors $\boldsymbol{p}_k$ of the contour. We now replace a Fourier series development by the discrete Fourier transform. Then the interpolated contour can be expressed in the Euclidean space as

$$C(\phi) \;=\; \sum_{k=-N}^{N} \boldsymbol{R}_k^{\phi} \, \boldsymbol{p}_k \, \widetilde{\boldsymbol{R}}_k^{\phi}.$$

For each ϕ does $C(\phi)$ lead to a point in the Euclidean space. We first have to transform this expression to conformal space. Then we can substitute this expression into the constraint equations for the pose estimation. The transformation of the Fourier descriptors in the conformal space can be expressed as

$$\mathbf{e} \wedge (C(\phi) + \mathbf{e}_-) \;=\; \mathbf{e} \wedge \left(\left(\sum_{k=-N}^{N} \boldsymbol{R}_k^{\phi} \, \boldsymbol{p}_k \, \widetilde{\boldsymbol{R}}_k^{\phi} \right) + \mathbf{e}_- \right).$$

The innermost parenthesis contains the Fourier descriptors in Euclidean space. The next parenthesis transform this expression to homogeneous space and then it is transformed to conformal space. Substituting this expression into the pose constraint equation leads to

$$\begin{aligned} \left(\boldsymbol{M}(\mathbf{e} \wedge (C(\phi) + \mathbf{e}_-))\widetilde{\boldsymbol{M}}\right) \;\underline{\times}\; (\mathbf{e} \wedge (\boldsymbol{O} \wedge \boldsymbol{x})) &= 0 \Leftrightarrow \\ \left(\boldsymbol{M} \left(\mathbf{e} \wedge \left(\left(\sum_{k=-N}^{N} \boldsymbol{R}_k^{\phi} \, \boldsymbol{p}_k \, \widetilde{\boldsymbol{R}}_k^{\phi} \right) + \mathbf{e}_- \right) \right) \widetilde{\boldsymbol{M}} \right) & \\ \underline{\times}\; (\mathbf{e} \wedge (\boldsymbol{O} \wedge \boldsymbol{x})) &= 0. \end{aligned}$$

The interpretation of this equation is simple: The innermost part contains the substituted Fourier descriptors in the conformal space. This is then coupled with the unknown rigid body motion (the motor $\boldsymbol{M}$) and compared with a reconstructed projection ray, also given in the conformal space.

Solving a set of constraint equations for a free-form contour with respect to the unknown motor $\boldsymbol{M}$ is a non-trivial task, since a motor corresponds to a polynomial of infinite degree. In [Rosenhahn et al 2002] we presented a method which does not estimate the rigid body motion on the Lie group $SE(3)$, but the parameters which generate their Lie algebra $se(3)$ (*twist approach*), comparable to the ideas, presented in [Bregler et al 1998; Lowe 1987]. Note: though the equations are expressed in a linear manner with respect to the group action, the equations in the unknown generators of the group action are non-linear and in the twist approach they will be linearized and iterated. This corresponds to a gradient descend method in the 3D space.

4 Experiments

In this section we present experimental results of free-form contour pose estimation. Therefore we will start with an introduction to the main algorithm for pose estimation of free-form contours. Though the numerical estimation of the pose parameters is already clarified in the last section, the main problem is to determine suited correspondences between 2D image features and points on the 3D model curve. Therefore a version of an ICP-algorithm is presented. To deal with 3D objects and partial occluded aspects of objects during tracking, we then present a modified version of our algorithm. There we are able to deal with occlusion problems by using sets of Fourier descriptors to model aspects of the object within different scenarios. The experiments conclude with modeling object deformations by combining kinematic chains within the Fourier descriptors for object modeling.

Using our approach for pose estimation of point-line correspondences, the algorithm for free-form contours consists of iterating the following steps:

```
(a) Reconstruct projection rays from the image points.
(b) Estimate the nearest point of each projection ray
    to a point on the 3D contour.
(c) Estimate the pose of the contour with the use of this
    correspondence set.
(d) goto (b).
```

The idea is, that all image contour points simultaneously pull on the 3D contour. The algorithm itself corresponds to the well-known ICP algorithm, e.g. discussed in [Rusinkiewicz et al 2001; Zang 1999]. But whereas it is mostly applied on sets of 2D or 3D points we apply it on a trigonometric interpolated function and from image points reconstructed projection rays. Note that this algorithm only works if we assume a scenario where the observations in the image plane are not too different. Thus, it is useful for tracking tasks. Further the monotonous convergence does sometimes lead to local minima. To avoid such local minima we also use low-pass information for contour approximation during the iteration. A projection of the used object model for our first experiments is shown in figure 4. The discrete points and different approximation levels are shown. The

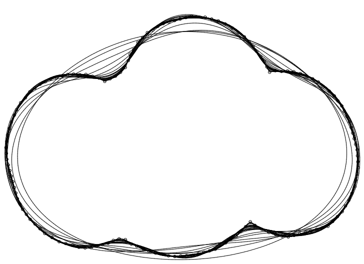

Figure 4: The different approximation levels of the 3D object contour.

principle of the ICP-algorithm during tracking is visualized in figure 5. There the chosen correspondences are also visualized by projecting them into the image. Note, that the correspondences are chosen in the 3D space, since the 3D contour points are related to 3D projection rays.

Figure 6 shows the computing times for an image sequence containing 520 images. The computing time for each image varies between 20ms and 55ms. The average computing time is 34ms, which is equivalent to 29 fps. These results were achieved with a 2GHz Pentium 4 computer.

Simultaneous pose estimation of multiple contours

In the previous experiment our object model is assumed as one (closed) contour. But many 3D objects can more easily be represented by a set of 3D contours expressing the different aspects of the object. We will now extend the object model to a set of 3D contours. The main problem is, how to deal with occluded or partially occluded contour parts of the

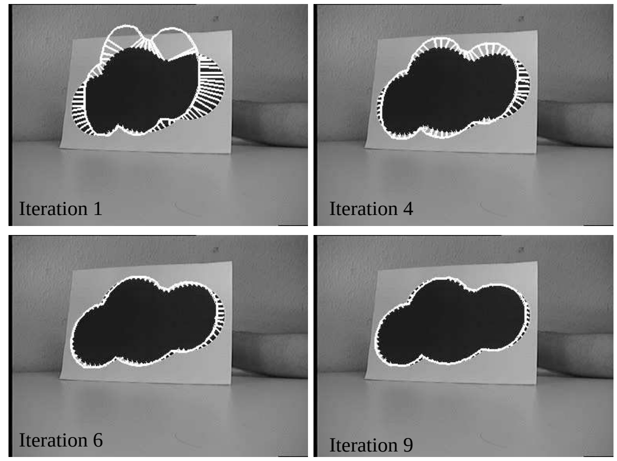

Figure 5: visualization of the ICP-algorithm during the iteration.

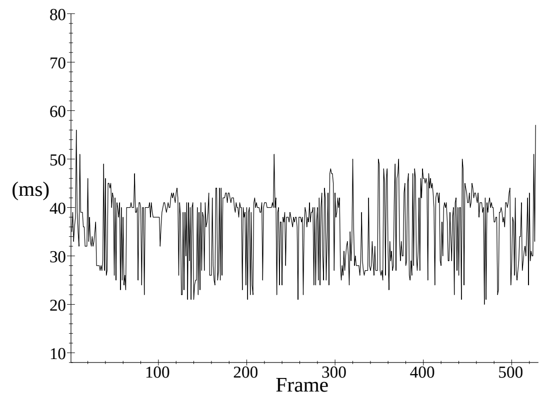

Figure 6: Computing times for an image sequence containing 520 images.

object. Here, the contours are assumed as rigidly coupled to each other. This means that the pose of one contour automatically defines the pose of the other

Our algorithm to deal with partially occluded object parts is simple and effective:

```
Assumptions: n 3D contours, one boundary contour in the image,
             dist(P,R) is a distance  function between a
             3D point P and a 3D ray R.
Result: Correspondences and pose.

(a) Reconstruct projection rays from the image points.
(b) For each projection ray R:
(c)   For each 3D contour:
     (c1) Estimate the nearest point P1 of ray R to a
          point on the contour.
     (c2) if (n==1) choose P1 as actual P for the
               point-line correspondence
     (c3) else compare P1 with P:
             if dist(P1,R) is smaller than dist(P,R)
               then choose P1 as new P
(d) Use (P,R) as correspondence set.
(e) Estimate pose with this correspondence set
(f) Transform contours, goto (b)
```

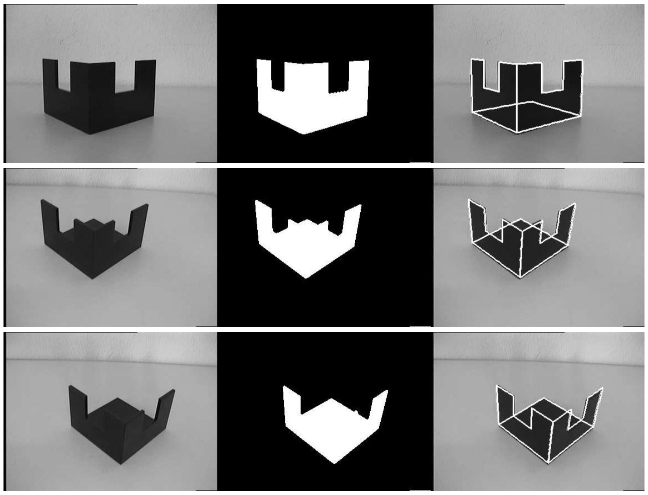

Figure 7: Pose results of an object with partially occluded contours. The left image shows the original image. The middle image shows the extracted silhouette and the right image visualizes the pose result.

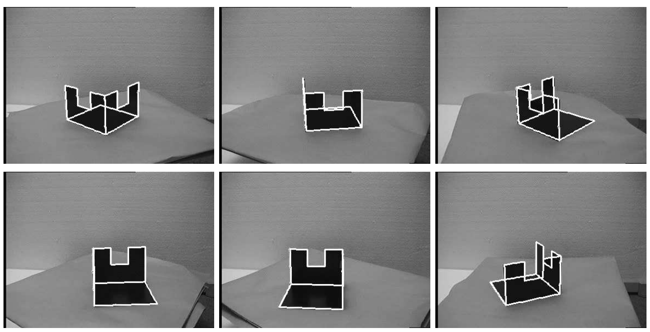

Figure 8: Pose results of an image sequence containing different aspect changes and degenerate situations.

The idea is to apply our ICP-algorithm not to one image contour and one 3D contour, but now to one image contour and a set of 3D contours.

This implies that for each extracted image point must exist one model contour and one point on this contour, which corresponds to this image point. Note, that the reverse is in general not the case.

Figure 7 visualizes the problem of partially occluded contour points. The only image information we use is the observed boundary contour of the object. By using a priori knowledge (e.g. assuming a tracking assumption), the pose can be recovered uniquely. This means, our algorithm can infer the position of hidden components from the visible components.

Our algorithm can even deal with aspect changes of the contour in an efficient manner. This is demonstrated in figure 8 in case of quiet different aspects of a 3D object. The images are taken from an image sequence containing 325 images. In this image sequence we put the object on a turn table and rotate it 360^o degree. Because the aspects of the object is changing, half-side models can no longer be used. Our tracking algorithm does not fail and is even able to cope with degenerate situations.

In another experiment, we use as object model the shape of a 3D tree. The contour approximations and pose results are shown in figure 9. The interesting part of this model, in contrast to the previous ones, is that it contains two nested contours. Consequently, it is much more complicated than the previous models.

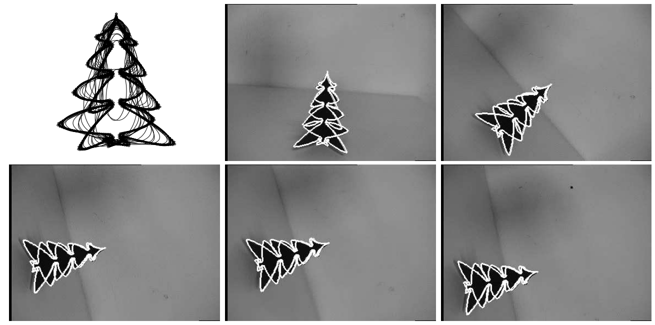

Figure 9: Approximations of the tree model and pose results of the tree model during an image sequence.

Modeling deformable objects

In the last experiment we want to report on our results in modeling slight *object deformations* during tracking. The idea is to embed an additional deformation function $\mathcal{D}$ within the pose constraints. To obtain such deformations, we couple the Fourier descriptors with kinematic chains. This is visualized in the right example of figure 10.

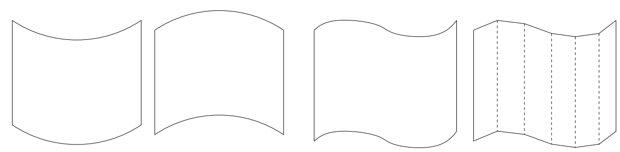

Figure 10: Possible deformation of a sheet of paper along the y-axes and their representation as kinematic chain.

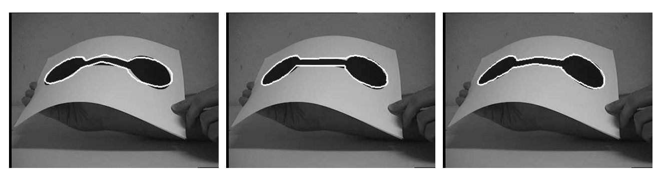

Figure 11: Pose result of a free-form object containing one, two or three kinematic chain segments.

Figure 11 visualizes the results of the algorithm for different numbers of used twists and shows that not many twists are needed to get a good approximation of the deformation. In our experiments we use 3-5 twists. Figure 12 shows pose results of an image sequence and visualizes the pose quality. As can be seen, the quality is very good, but more important is the fact, that the computing time does not increase too much. In our actual version we need a computing time of 70 ms for each image on a Linux 2 GHz machine, to estimate the pose and deformation angles.

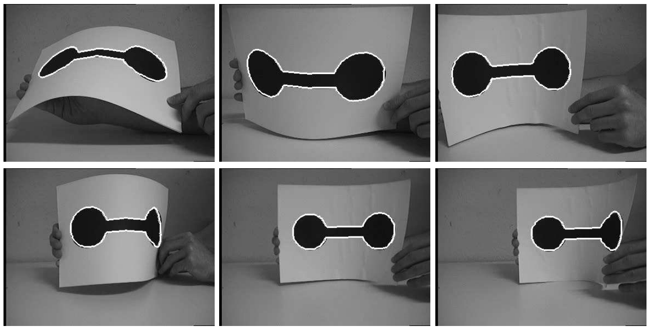

Figure 12: Pose results of a free-form object taken from an image sequence with 520 images.

5 Discussion

This work concerns the problem of 2D-3D pose estimation of 3D free-form contours. We assume the knowledge of a 3D object, which contains one or more contours modelling the aspects of the object. Furthermore, we assume a calibrated camera and observe the silhouette of the object in the camera. The aim is to estimate the pose, which leads to a best fit between image and model data. We explain how to estimate 3D closed contours by using 3D Fourier descriptors and use the conformal geometric algebra to compare the 3D contours with (from image points) reconstructed projection rays. This leads to constraint equations which are solved by using a gradient descend method, combined with an ICP-algorithm. The experiments show the efficiency of our algorithm on different image sequences. We further deal with partially occluded object features and nested contours and show, that even extensions like the modeling of object deformations by combining Fourier descriptors with kinematic chains is possible in this framework.

Acknowledgements

This work has been supported by DFG Graduiertenkolleg No. 357 and by EC Grant IST-2001-3422 (VISATEC).

References

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.

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.

Drummond T. and Cipolla R. Real-time tracking of multiple articulated structures in multiple views. In *6th European Conference on Computer Vision, ECCV 2000, Dubline, Ireland*, Part II, pp.20-36, 2000.

Granlund G. Fourier preprocessing for hand print character recognition. *IEEE Transactions on Computers*, Vol. 21, pp. 195-201, 1972.

Grimson W. E. L. Object Recognition by Computer. *The MIT Press, Cambridge, MA*, 1990.

Lowe D.G. Solving for the parameters of object models from image descriptions. in *Proc. ARPA Image Understanding Workshop*, pp. 121-127, 1980.

Lowe D.G. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, Vol. 31, No. 3, pp. 355-395, 1987.

Kriegman D.J., Vijayakumar B., and Ponce, J. Constraints for recognizing and locating curved 3D objects from monocular image features. In *Proceedings of Computer Vision (ECCV '92), G. Sandini, (Ed.)*, LNCS 588, Springer-Verlag, pp. 829–833, 1992

Li H., Hestenes D. and Rockwood A. Generalized homogeneous coordinates for computational geometry. In [Sommer 2001], pp. 27-52, 2001.

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

Needham T. Visual Complex Analysis. *Oxford University Press*, 1997

Reiss T.H. Recognizing Planar Objects Using Invariant Image Features. LNCS 676, *Springer Verlag*, 1993.

Rosenhahn B., Zhang Y. and Sommer G. Pose estimation in the language of kinematics. In: *Second International Workshop, Algebraic Frames for the Perception-Action Cycle, AFPAC 2000, Sommer G. and Zeevi Y.Y. (Eds.)* , LNCS 1888, Springer-Verlag, Heidelberg, pp.284-293, 2000.

Rosenhahn B., Granert O., Sommer G. Monocular pose estimation of kinematic chains. In *Applied Geometric Algebras for Computer Science and Engineering, Birkhäuser Verlag, Dorst L., Doran C. and Lasenby J. (Eds.)*, pp. 373-383, 2002.

Rosenhahn B. and Sommer G. Adaptive Pose Estimation for Different Corresponding Entities. In *Pattern Recognition, 24th DAGM Symposium, L. Van Gool (Ed.)*, Springer-Verlag, Berling Heidelberg, LNCS 2449, pp. 265-273, 2002.

Rusinkiewicz S. and Levoy M. Efficient variants of the ICP algorithm. Available at http://www.cs.princeton.edu/ smr/papers/fasticp/. Presented at *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001.

Sommer G., editor. Geometric Computing with Clifford Algebra. *Springer Verlag*, 2001.

Stark K. A method for tracking the pose of known 3D objects based on an active contour model. *Technical Report TUD / FI 96 10*, TU Dresden, 1996.

Zang Z. Iterative point matching for registration of free-form curves and surfaces. *IJCV: International Journal of Computer Vision*, Vol. 13, No. 2, pp. 119-152, 1999.

Zerroug, M. and Nevatia, R. Pose estimation of multi-part curved objects. *Image Understanding Workshop (IUW)*, pp. 831-835, 1996

Zahn C.T. and Roskies R.Z. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, Vol. 21, No. 3, pp. 269-281, 1972.