Modeling Adaptive Deformations during Free-form Pose Estimation

Bodo Rosenhahn, Christian Perwass, 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 deformable 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 of the 3D object to the reference camera system. The object itself is modeled as free-form contour. 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 unique entities with 3D Fourier descriptors and combined with an ICP (Iterative Closest Point) algorithm they are embedded in the pose problem. The modeling of object deformations within free-form pose estimation is achieved by a combination of adaptive kinematic chain segments within Fourier descriptors.
Keywords: Pose estimation, Fourier descriptors, kinematic chains

1 Introduction

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. Pose estimation means to estimate the relative position and orientation of the 3D object to the reference camera system: We assume a 3D object model and 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. Pioneering work was done in the 80's and 90's by Lowe [6], Grimson [5] and others. In their work, point correspondences are used. More abstract entities can be found in [15, 2]. In the literature we find circles, cylinders, kinematic chains or other multi-part curved objects as entities. Works concerning free-form curves can be found in [3, 13]. Contour point sets, affine snakes, or active contours are used for visual servoing in these works.

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 through projective reconstruction from the image entity. This idea will be used to formulate the scenario as a pure 3D problem. Our recent work [11] concentrates on modeling objects by using features of the object (e.g. corners, edges). Instead, we now deal with 3D contours of the object. 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. In such cases there is need to deal for example with the silhouette of the object as a whole, instead of with sparse local features of the silhouette. In these scenarios free-form contours are applied. The motivation for modeling object deformations within free-form contours is shown in figure 1. In

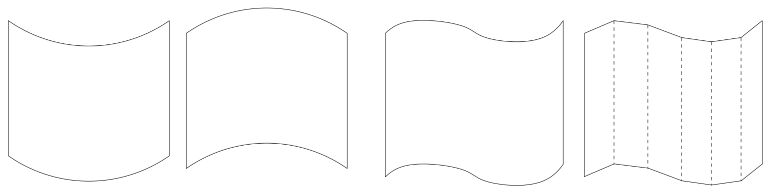

Fig. 1. Possible deformations of a sheet of paper along the y-axis and their representation as kinematic chain.

our experiments we use a planar object, which is printed on a sheet of paper. To model deformations of the sheet of paper we combine kinematic chains within the object contour as shown in the right image of figure 1. Another main point in this contribution is additionally to model object deformations in an adaptive manner: Kinematic chains are used within free-form contours. But the scenario during an image sequence may change, so that it is not useful to take a fixed number of joints along the kinematic chain. Instead, we present a real-time system, which chooses the number of joints adaptively. This leads to more stable and time-optimized algorithms.

2 The pose problem in conformal geometric algebra

This section concerns the formalization of the free-form pose estimation 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 possibility of coupling projective, kinematic and Euclidean geometry by using a conformal model and its dense symbolic representation. A more detailed introduction concerning geometric algebras can be found in [12].

The main idea of geometric algebras $\boldsymbol{\mathcal{G}}$ 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 $\boldsymbol{\mathcal{G}}_{p,q}$ 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

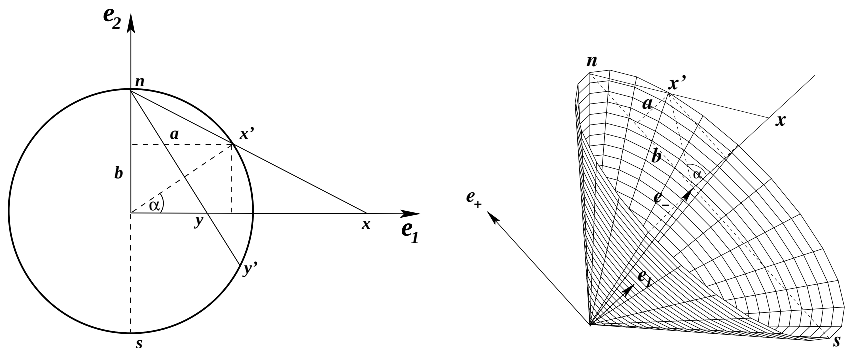

Fig. 2. Left: Visualization of a stereographic projection for the 1D case: Points on the line $\mathbf{e}_1$ are projected on the (unit) circle. Right: Visualization of the homogeneous model for a stereographic projection in the 1D case. All stereographic projected points are on a cone, which is a null-cone in the Minkowski space.

anticommutator $\overline{\times}$ product. The idea behind conformal geometry is to interpret points as *stereographically projected* points. This means augmenting the dimension of space by one. The method used in a stereographic projection is visualized for the 1D case in the left image of figure 2: Points $\boldsymbol{x}$ on the line $\mathbf{e}_1$ are mapped to points $\boldsymbol{x}'$ on the unit circle by intersecting the line spanned by the *north pole* $\boldsymbol{n}$ and $\boldsymbol{x}$ with the circle. The basic formulas for projecting points in space on the sphere and vice versa are for example given in [9]. Using a homogeneous model for stereographic projected points means to augment the coordinate system by a further additional coordinate whose unit vector now squares to minus one. In 1D this leads to a cone in space, which is visualized in the right image of figure 2. This cone is spanned by the original coordinate system, the augmented one of the stereographic projection and the homogeneous one. This space introduces a Minkowski metric and will lead to a representation of any Euclidean point on a null-cone (1D case) or a hyper-null-cone (3D case). In [12] it is further shown that the conformal group of $\mathbb{R}^n$ is isomorphic to the Lorentz group of $\mathbb{R}^{n+1,1}$ which has a spinor representation in $\mathcal{G}_{n+1,1}$. We will take advantage of both properties of the constructed embedding which are the representation of points as null-vectors and the spinor representation of the conformal group.

The conformal geometric algebra $\mathcal{G}_{4,1}$ (CGA) [7] is suited to describe conformal geometry. It contains spheres as geometric basis entities and the conformal transformations as geometric manipulations. The point at infinity, $\mathbf{e} \simeq \boldsymbol{n}$, and the origin, $\mathbf{e}_0 \simeq \boldsymbol{s}$, are special elements of the representation which are used as basis vectors instead of $\mathbf{e}_+$ and $\mathbf{e}_-$ because they define a null space in the conformal geometric algebra. A Euclidean point $\boldsymbol{x} \in \mathbb{R}^3$ can be represented as a point $\underline{\boldsymbol{x}}$ on the null-cone by taking $\underline{\boldsymbol{x}} = \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathbf{e} + \mathbf{e}_0$. 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 $\underline{\boldsymbol{s}} = \boldsymbol{p} + \frac{1}{2}(\boldsymbol{p}^2 - \rho^2)\mathbf{e} + \mathbf{e}_0$, 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 define entities like points, lines, planes or circles as subspaces, generated from spheres.

Rotations are represented by rotors, $\boldsymbol{R} = \exp\left(-\frac{\theta}{2}\boldsymbol{l}\right)$. The parameter of a rotor $\boldsymbol{R}$ is the rotation angle θ applied on a unit bivector $\boldsymbol{l}$ which represents the dual of the rotation axis. 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, $\boldsymbol{T} = (1 + \frac{\mathbf{e}\boldsymbol{t}}{2}) = \exp\left(\frac{\mathbf{e}\boldsymbol{t}}{2}\right)$. A rigid body motion can be expressed by a screw motion [8]. For every screw motion $g \in SE(3)$ 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 screw motion has the general form $\boldsymbol{M} = \exp(-\frac{\theta}{2}(\boldsymbol{n}+\mathbf{e}\boldsymbol{m}))$, with a unit bivector $\boldsymbol{n}$ and an arbitrary 3D vector $\boldsymbol{m}$. The triple $(\theta, \boldsymbol{n}, \boldsymbol{m})$ in the exponential term represent the twist parameters. Whereas in Euclidean geometry, Lie groups and Lie algebras 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 start to express the 2D-3D pose estimation problem for pure point correspondences: *a transformed object point has to lie on a projection ray, reconstructed from an image point.* Let $\underline{\boldsymbol{X}}$ be a 3D object point given in CGA. The (unknown) transformation of the point can be described as $\boldsymbol{M}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{M}}$. Let $\boldsymbol{x}$ be an image point 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 line $\underline{\boldsymbol{L}}_x$ is calculated from the optical center $\boldsymbol{O}$, the image point $\boldsymbol{x}$ and the vector $\mathbf{e}$ as the point at infinity. The line $\underline{\boldsymbol{L}}_x$ is given in a Plücker representation. Collinearity can be described by the commutator product. Thus, the 2D-3D pose estimation from an image point can be formalized as constraint equation in CGA,

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

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.

Fourier descriptors in CGA

Fourier descriptors are often used for object recognition [4] and affine pose estimation [1] of closed contours. We are now concerned with the formalization of 3D Fourier descriptors in CGA in order to combine these with our previously introduced pose estimation constraints. Let $\boldsymbol{R}_i^{\phi} := \exp(-\pi u_i \phi/T)\,\boldsymbol{l})$, 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, $\widetilde{\boldsymbol{R}}_i^{\phi} = \exp(\pi u_i \phi/T\,\boldsymbol{l})$. Because $\boldsymbol{l}^2 = -1$ we can write the exponential function as $\exp(\phi\boldsymbol{l}) = \cos(\phi) + \boldsymbol{l}\sin(\phi)$. We can now formulate any closed curve $C(\phi)$ of the Euclidean plane 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 plane curve there is a unique set of phase vectors $\{\boldsymbol{p}_k\}$ that parameterize the curve. To represent a general closed, discretizied 3D curve this can easily be extended to 3D by interpreting the projections along x, y, and z as three infinite 1D-signals and applying a DFT and an IDFT separately, leads to the representation

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

This means we now replace a Fourier series development by the inverse discrete Fourier transform.

Pose estimation of free-form contours

We assume a given closed, discretizied 3D curve, that is a 3D contour C with $2N+1$ sampled points in both the spatial and spectral domain with phase vectors $\boldsymbol{p}_k^m$ of the contour. Substituting the representation of the Fourier descriptors in the conformal space within the pose estimation constraint equations leads to

$$\left(\boldsymbol{M}\left(\mathbf{e}\wedge(C(\phi)+\mathbf{e}_-)\right)\widetilde{\boldsymbol{M}}\right)\ \underline{\times}\ (\mathbf{e}\wedge(\boldsymbol{O}\wedge\boldsymbol{x}))=0.$$

To model any additional deformation, a kinematic chain is added within the pose constraint. This means encapsulating n motors $\boldsymbol{M}_i$ of the deformations within the constraint equation,

$$\left(\boldsymbol{M}\left(\prod_{i=1}^{n}\boldsymbol{M}_i^{\theta_i}\left(\mathbf{e}\wedge(C(\phi)+\mathbf{e}_-)\right)\widetilde{\boldsymbol{M}}_i^{\theta_i}\right)\widetilde{\boldsymbol{M}}\right)\ \underline{\times}\ (\mathbf{e}\wedge(\boldsymbol{O}\wedge\boldsymbol{x}))=0.$$

This constraint equation is easy to interpret: The inner parenthesis contains the inverse Fourier transformed phase vectors transformed to a representation in the conformal space. The next parenthesis contains the motors $\boldsymbol{M}_i^{\theta_i}$, which are exponentials of twists modeling the joints of the kinematic chain. The last parenthesis contains the motor $\boldsymbol{M}$ with the unknown pose. This is then coupled with the reconstructed projection ray in the conformal space. The unknowns are the pose parameters $\boldsymbol{M}$, the angles θ_i of the kinematic chain and the angle ϕ of the Fourier descriptors.

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 [10] 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, $se(3)$, comparable to the ideas, presented in [2, 6]. Note, that 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 our approach they will be linearized and iterated. This corresponds to a gradient descent method in 3D space.

3 Experiments

In this section we present experimental results of free-form pose estimation. The algorithm for deformable free-form pose estimation is basically a modified ICP-algorithm [14]. The convergence behavior of the ICP-algorithm during an image sequence is shown in figure 3. As can be seen, we refine the pose results by using

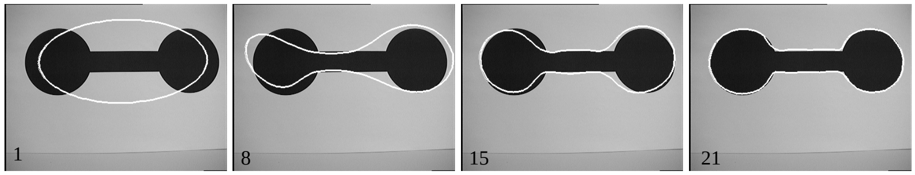

Fig. 3. Pose results of the low-pass filtered contour during the iteration.

a low-pass approximation for pose estimation and by adding successively higher frequencies during the iteration. This is basically a multi-resolution method and helps to avoid local minima during the iteration.

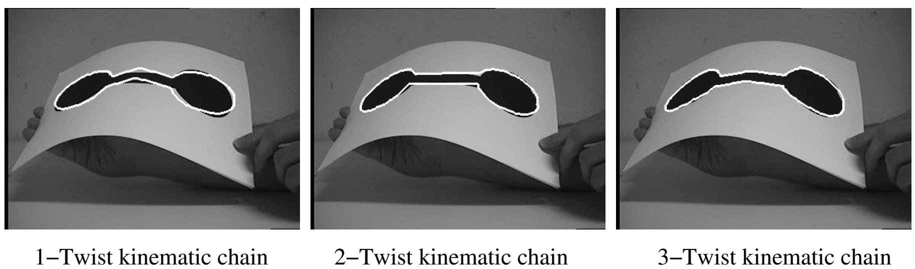

Fig. 4. Pose result of a free-form contour containing one, two or three joints.

To model object deformations, the effect of introducing different numbers of joints (as twists) within the pose scenario is visualized in figure 4. It can be seen, that only a few twists are needed to get a good approximation of the deformation. There are two major problems in dealing with a fixed set of twists modeling the object deformation: Firstly, the use of too many twists can lead to local minima and wrong poses. This occurs for example in case of using too many twists for modeling only a slight object deformation. Secondly does the use of many twists increase the computing time of the pose estimation algorithm, since additionally unknowns are modeled which are not always needed. Therefore, a modification of the algorithm is done which chooses the number of twists adaptively, depending on the level of deformation. An example of an image sequence is shown in figure 5. The diagram shows the frame number during the image sequence on the x-axis and the used number of twists on the y-axis. The example images show that the used number of twists is consistent with the degree of deformation. The increased time performance is shown in figure 6. The y-axis shows on the one hand the used number of twists (consistent with figure

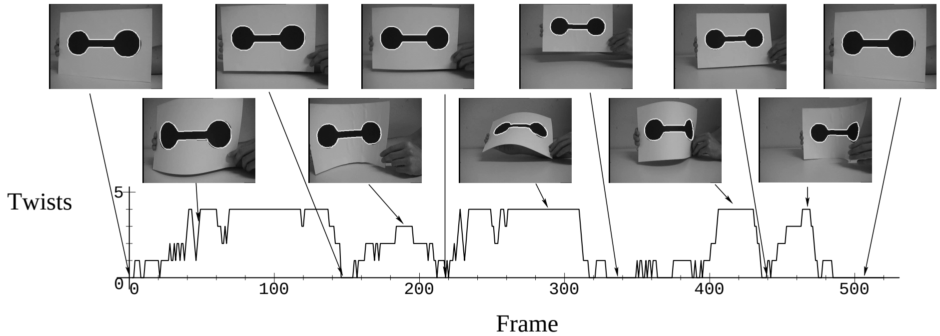

Fig. 5. Adaptive choice of twists for modeling object deformations during an image sequence. For slight deformations less twists are used then for larger ones.

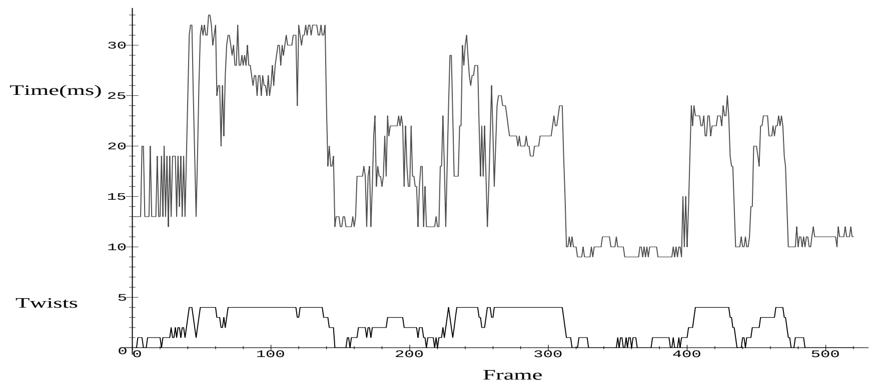

Fig. 6. Time performance for using different numbers of twists in an image sequence

5) and on the other hand the computing time for estimating one pose during the ICP-algorithm. As can be seen, the use of more twists increases the computing time, and the adaptive choice of the number of twists during the image sequence leads to a situation dependent optimized time performance. Note that only the time for estimating one pose is shown. Combined with the ICP-algorithm (which takes between two and eight iteration steps), the overall computing time for one frame varies between $20ms$ and $250ms$. The adaptive behavior is achieved by evaluating the twist angles after each processed image. If the angles are below or above a threshold, twists are eliminated or added and rearranged, respectively. The experiments are performed with a Linux 2 GHz machine.

4 Discussion

This work presents a novel approach to deal with plane dynamic deformations of 3D free-form curves during pose estimation. Free-form contours are modeled by 3D Fourier descriptors which are combined with pose estimation constraints. This coupling of geometry with signal theory is achieved by using the conformal geometric algebra. In this language we are able to fuse concepts, like complex numbers, Plücker lines, twists, Lie algebras and Lie groups in a compact manner. The chosen framework shows, that it is possible to extend scenarios to more complex ones, without loosing the geometric oversight since the equations are given in closed and easily interpretable forms. Our future work will concentrate on dealing with more complex scenarios, e.g. the modeling of free-form surfaces.

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

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. 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.
4. Granlund G. Fourier preprocessing for hand print character recognition. *IEEE Transactions on Computers*, Vol. 21, pp. 195-201, 1972.
5. Grimson W. E. L. Object Recognition by Computer. *The MIT Press, Cambridge, MA*, 1990.
6. Lowe D.G. Solving for the parameters of object models from image descriptions. in *Proc. ARPA Image Understanding Workshop*, pp. 121-127, 1980.
7. Li H., Hestenes D. and Rockwood A. Generalized homogeneous coordinates for computational geometry. In [12], pp. 27-52, 2001.
8. Murray R.M., Li Z. and Sastry S.S. A Mathematical Introduction to Robotic Manipulation. *CRC Press*, 1994.
9. Needham T. Visual Complex Analysis. *Oxford University Press*, 1997
10. Rosenhahn B., Perwass Ch. and Sommer G. Pose estimation of 3D free-form contours in conformal geometry In *Proceedings of Image and Vision Computing (IVCNZ) D. Kenwright (Ed.)*, New Zealand, pp. 29-34, 2002.
11. 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.
12. Sommer G., editor. Geometric Computing with Clifford Algebra. *Springer Verlag*, 2001.
13. 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.
14. 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.
15. Zerroug, M. and Nevatia, R. Pose estimation of multi-part curved objects. *Image Understanding Workshop (IUW)*, pp. 831-835, 1996