Pose Estimation of Free-Form Objects

Bodo Rosenhahn[1] and Gerald Sommer[2]

[1] University of Auckland (CITR)
Computer Science Department
Tamaki Campus
Private Bag 92019 Auckland
New Zealand
bros028@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 approach for 2D-3D pose estimation of 3D free-form surface models. In our scenario we observe a free-form object 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 a two-parametric 3D surface and extended by one-parametric contour parts of the object. A twist representation, which is equivalent to a Fourier representation allows for a low-pass approximation of the object model, which is advantageously applied to regularize the pose problem. The experiments show, that our developed algorithms are fast (200ms/frame) and accurate (1^o rotational error/frame).

1 Introduction

Pose estimation itself is one of the oldest computer vision problems. It is crucial for many computer and robot vision tasks. Pioneering work was done in the 80's and 90's by Lowe [8], Grimson [7] and others. These authors use point correspondences. More abstract entities can be found in [16,3]. 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 e.g. in [5]. Contour point sets, affine snakes, or active contours are used for visual servoing in different works. For a definition of the pose problem we want to quote Grimson [7]: *By pose we mean the transformation needed to map an object model from its inherent coordinate system into agreement with the sensory data.* We are estimating the relative rotation and translation of a 3D object with respect to a reference camera system in the framework of a 2D-3D pose estimation approach. In this work we deal with free-form surface and contour models for object representation. We want to quote Besl [2] for a definition: *A free-form surface has a well defined surface that is continuous almost everywhere except at vertices, edges and cusps.*

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3021, pp. 414–427, 2004.

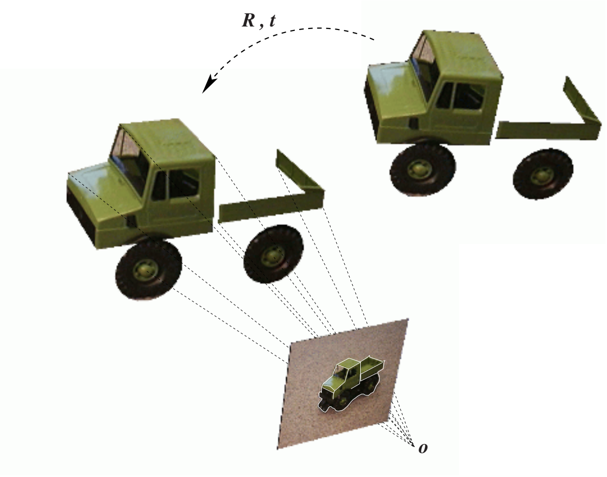


Fig. 1. The scenario. The assumptions are the projective camera model, the model of the object, extracted features and the silhouette of the object on the image plane. The aim is to find the pose ($\boldsymbol{R}$, $\boldsymbol{t}$) of the model, which leads to the best fit of the object with the image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

In section 2 we start with a summary of our preliminary works regarding feature based and contour based pose estimation [12]. Then we present our extensions in section 3: An approach for silhouette based pose estimation of free-form surface models. In this approach we assume an extracted image contour of the observed object model. Only by using this contour we estimate the pose of the surface model with respect to a calibrated reference camera system. The surface model is parametrically represented based on a signal model from which low-pass approximations are derived.

It is clear, that only using the image contour results in a loss of further available object information in the image plane. Therefore, we will also present an extension which takes into account additionally available object information. We are using additional contour parts of the object which are brought to correspondence with extracted image features inside the object silhouette.

In the experiments, section 4, we present pose results of objects which are tracked successfully, even with noisy extracted image contours. We will show that our algorithms are able to cope with occlusions caused by the motion and to compensate errors to some degree. The contribution ends with a discussion in section 5.

To deal with geometric aspects of the pose problem, we use as mathematical language so-called Clifford or geometric algebras [14]. Here we will give no theoretical introduction into the concepts of Clifford algebras but want to point out a few properties which are important for this problem: The elements in geometric algebras are called multivectors which can be multiplied by using a geometric product. In geometric algebra Euclidean, projective and conformal geometry [9] find the frame where they can reconcile and express their respective potential. Besides, it enables a coordinate-free and dense symbolic representation. To model the pose problem, we use the conformal geometric algebra (CGA). The CGA is build up on a conformal model (geometry on the sphere) which is coupled with a homogeneous model to deal with kinematics and projective geometry simultaneously. This enables us to deal with the Euclidean, kinematic and projective space in one framework and therefore to cope with the pose problem in an efficient manner. Furthermore the unknown rigid motions are expressed as so-called *motors* which can be applied on different entities (e.g. points or lines) by the use of the geometric product. This leads to compact and easily interpretable equations. 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 [12] for more detailed information.

2 Preliminary Work

We start with a few aspects of our preliminary works which build the basis for this contribution. First, we will present point based pose estimation and then we continue with the approach for contour based free-form pose estimation.

2.1 Point Based Pose Estimation

For 2D-3D point based pose estimation we are using constraint equations which compare 2D image points with 3D object points. The use of points is the simplest representation for 3D objects treated here. To compare a 2D image point $\boldsymbol{x}$ with 3D object points $\underline{\boldsymbol{X}}$, the idea is to reconstruct from the image point a 3D projection ray, $\underline{\boldsymbol{L}}_x = \mathbf{e} \wedge (\boldsymbol{O} \wedge \boldsymbol{x})$, as Plücker line [10]. The motor $\boldsymbol{M}$ as exponential of a twist, Ψ, $\boldsymbol{M} = \exp(-\frac{\theta}{2}\Psi)$, formalizes the unknown rigid motion as a screw motion [10]. The motor $\boldsymbol{M}$ is applied on the 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. The representation of such a constraint equation takes in geometric algebra the form

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

Note, that we work with a 3D formalization of the pose problem. The constraint equations can be solved by linearization (this means solving the equations for the twist-parameters which generate the screw motion) and by applying the Rodrigues formula for reconstruction of the group action [10]. Iteration leads to a gradient descent method in 3D space. This is more detailed presented in [12].

There we also introduce similar equations to compare 3D points with 2D lines (3D planes) and 3D lines with 2D lines (3D planes). The pose estimation can be performed in real-time and we need 2ms to estimate a pose containing 100 point correspondences on a Linux 2GHz machine.

2.2 Contour Based Pose Estimation

Though point concepts or higher order features are often used for pose estimation [3,16], there exist certain scenarios (e.g. in natural environments), where it is not possible to extract features like corners or curve segments, but just general contours. Therefore we are interested in modeling free-form objects and embedding them into the pose problem.

Fourier descriptors can be used for object recognition [6] and affine invariant pose estimation [1] of closed contours. They have the advantage of a low-pass object representation (as explained later) and they interpolate sample points along a contour as a continuously differentiable function. During our research we rediscovered the use of Fourier descriptors since they are the generalization of so-called *twist-generated* curves we used to model cycloidal curves (cardioids, nephroids etc.) within the pose problem [12]. We now deal with the representation of 3D free-form contours in order to combine these with our previously introduced point based pose estimation constraints. Since the later introduced pose estimation algorithm for surface models goes back to a contour based one, the recapitulation of our former works on contour based pose estimation is of importance.

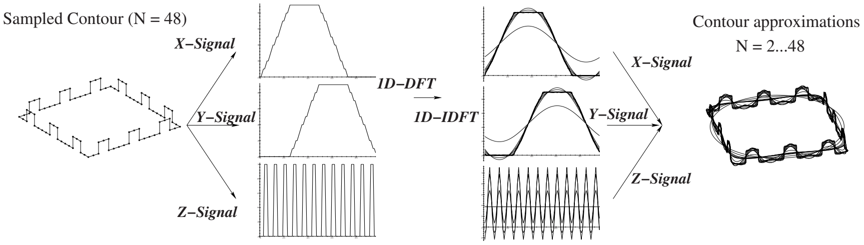


Fig. 2. Visualization of contour modeling and approximation by using three 1D Fourier transformations.

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 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

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

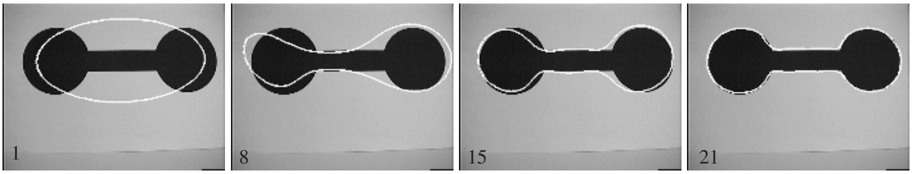


Fig. 3. Pose results of the low-pass filtered contour during the iteration.

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 $i = \sqrt{-1}$ with 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. The principle of modeling free-form contours is visualized in figure 2.

For pose estimation this model is then combined with a version of an ICP-algorithm [15]. Figure 3 shows an example. As can be seen, we refine the pose results by adding successively higher frequencies to a low-pass approximation during the iteration. This is basically a multi-resolution method and helps to avoid local minima during the iteration.

3 Surface Based Pose Estimation

After this recapitulation we will now present the main ideas for surface based pose estimation. We start with the extension of the 3D contour model to a 3D surface model and present the basic pose estimation algorithm for free-form surfaces [13]. Then we will continue with extensions of this approach.

3.1 Surface Representation

We are now concerned with the formalization of surfaces in the framework of 2D Fourier descriptors. This will enable us to regularize the estimation and to refine the object model during iteration steps. Hence the multi-scale object representation can be adapted to its inherent geometric complexity. We assume a two-parametric surface [4] of the form

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

This means, we have 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. Projecting this function along $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ leads to the three 2D functions $f^i(\phi_1, \phi_2)$. 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

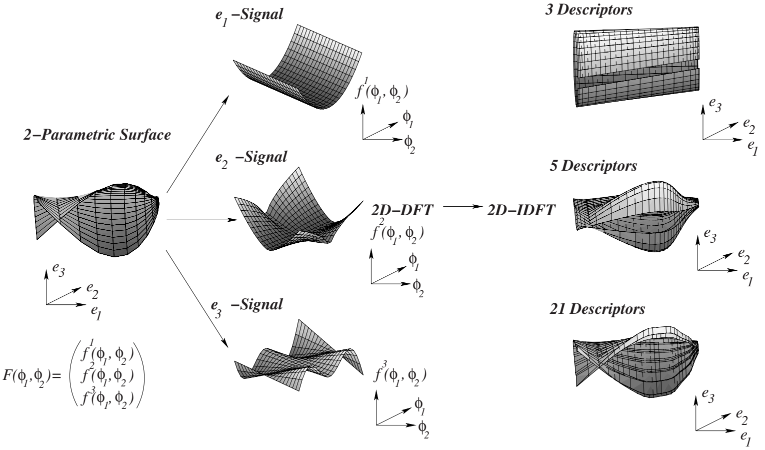


Fig. 4. Visualization of surface modeling and approximation by using three 2D Fourier transformations.

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 which appears in geometric algebra as

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

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,

$$\begin{aligned} \boldsymbol{p}^i_{k_1,k_2} &= \frac{1}{(2N_1+1)(2N_2+1)} \\ &\quad \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} f^i_{n_1,n_2} \exp\left(-\frac{2\pi k_1 n_1}{2N_1+1}\,\boldsymbol{l_i}\right) \exp\left(-\frac{2\pi k_2 n_2}{2N_2+1}\,\boldsymbol{l_i}\right)\mathbf{e}_i . \end{aligned}$$

This is visualized in figure 4 as extension to the 1D case of figure 2: a two-parametric surface can be interpreted as three separate 2D signals interpolated and approximated by using three 2D-DFTs and 2D-IDFTs, respectively.

3.2 Silhouette Based Pose Estimation of Free-Form Surfaces

We now continue with the algorithm for silhouette based pose estimation of surface models. In our scenario, we assume to have extracted the silhouette

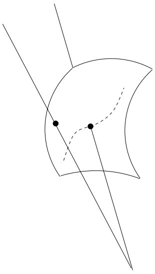


Fig. 5. A main problem during pose estimation of surface models: There is need to express tangentiality between the surface and the reconstructed projection rays. Pure intersection is not sufficient for pose estimation.

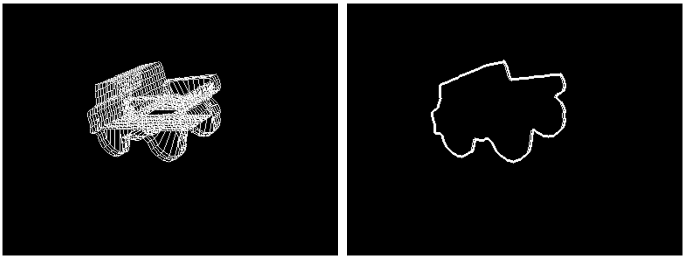
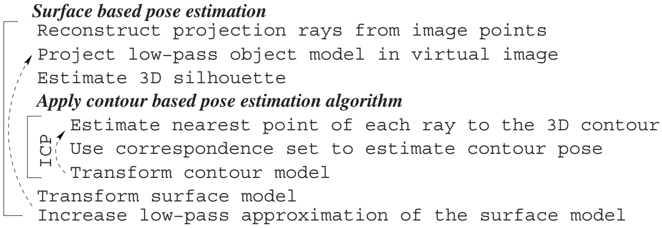


Fig. 6. Left: The surface model projected on a virtual image. Right: The estimated 3D silhouette of the surface model, back projected in an image.



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

Fig. 7. The algorithm for pose estimation of surface models.

of an object in an image. In the experiments this is simply done by using a thresholded color interval and by smoothing the resulting binary image with morphological operators, see figure 10.

A main problem is, that it is not useful to express an intersection constraint between the reconstructed projection rays and the surface model. This is visualized in figure 5: Postulating the intersection of rays with the surface leads to the effect, that the object is moved directly in front of the camera. Then every reconstructed ray intersects the surface and the constraint is trivially fulfilled. Therefore there is need to express tangentiality between the surface and the reconstructed projection rays and there is need to express a distance measure within our description.

To solve this problem we propose to get from the surface model to a contour model which is tangential with respect to the camera coordinate system. To compare points on the image silhouette with the surface model, the idea is to work with those points on the surface model which lie on the outline of a 2D projection 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. This is visualized in figure 6. The contour model is then applied on our previously introduced contour based pose estimation algorithm. Since the aspects of the surface model are changing during the 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 7.

Note, this approach can easily be extended to a multiple-component silhouette based pose estimation algorithm: If an object consists of several rigidly coupled surface patches, still one 3D contour can be estimated from the including free-form parts and applied to the pose estimation algorithm. This is presented in section 4.2.

3.3 Combining Contour and Surface Patches

We will now present a mixed-mode approach which applies additional edge information on the silhouette based pose estimation. We call additional edges, which are not on the outline of the surface contour with respect to the camera, 'internal' edges, since they are inside the boundary contour in the image. Depending on the object, they can be easily obtainable features which we want to use as additional information to stabilize the pose result. This means to extend the assumed model from one 3D component to multiple components of different dimension. These additional components are representing parts of contours within the outer silhouette of the object. To obtain 'internal' edge information we perform the following image processing steps:

```
1. Back-ground subtraction from the object
2. Laplace filtering and subtraction of the contour from
   the filtered image
3. Sub-sampling
```

This is visualized in figure 8: The first image shows the back-ground subtraction from the object. After this we filter the image and estimate an internal edge image as shown in the second image. The third image shows the sub-sampling to obtain a number of *internal* points we use for pose estimation.

It is useless to claim incidence of these extracted points with one given surface model since they do not contribute any information on the pose quality (as discussed in section 3.2). Instead, within the mixed-mode model of multiple components, their contribution to the pose estimation results increases the accuracy and the robustness with respect to occlusions. The generated set of equations

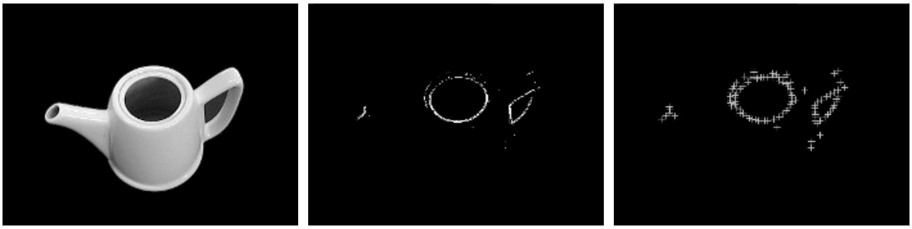


Fig. 8. Image processing steps for getting internal object features: Background subtraction and internal edge detection.

can now be separated in two parts, those obtained from the silhouette and those obtained from the internal feature points. Since both parts can contain larger mismatches or wrong correspondences (see e.g. the falsely extracted edges in figure 8), an outlier elimination is applied to reduce wrong correspondences [12].

4 Experiments

In the experiments we will start with results obtained from a pure silhouette based pose estimation of single patch surface models and continue with the use of multiple surface patches. In section 4.3 we will then deal with pose estimation using additional internal object features.

4.1 One-Component Silhouette Based Pose Estimation

The convergence behavior of the silhouette based pose estimation algorithm is shown in figure 9. As can be seen, we refine the pose results by adding successively higher frequencies during the iteration. This is basically a multi-resolution method and helps to avoid getting stuck in local minima during the iteration. The aim of the first experiment is not only to visualize the pose results, but also to compare the pose results with a ground truth: We put a car model on a turn table and perform a 360 degrees rotation. We further assume the Euclidean 3D surface model of the car and a calibrated camera system observing the turn table. The rotation on the turn table corresponds to a 360 degrees rotation around the y-axis in the calibrated camera system. During the image sequence we apply the silhouette based free-form pose estimation algorithm. Example images (and pose results) of this sequence with extracted image silhouettes are shown in figure 10. As can be seen, there exist shadows under the car, which lead to noisy segmented images. Also some parts of the car (e.g. the front bumper or the tow coupling) are not exactly modeled. This results in errors which are detected during pose estimation. After the detection of failure correspondences they are eliminated in the generated system of equations and they do not influence the result of the pose. Figure 11 shows the absolute error of the estimations in degrees during the whole image sequence. In the image sequence, the maximum error is 5.73 degrees (at the beginning of the sequence). The average absolute

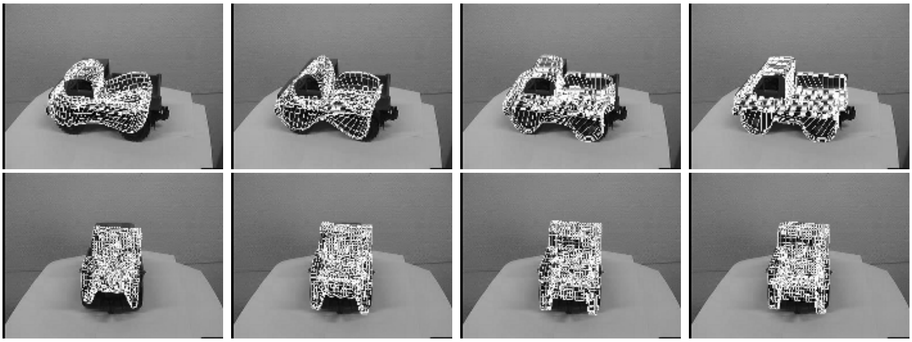


Fig. 9. Pose results of the low-pass contours during the ICP cycle.

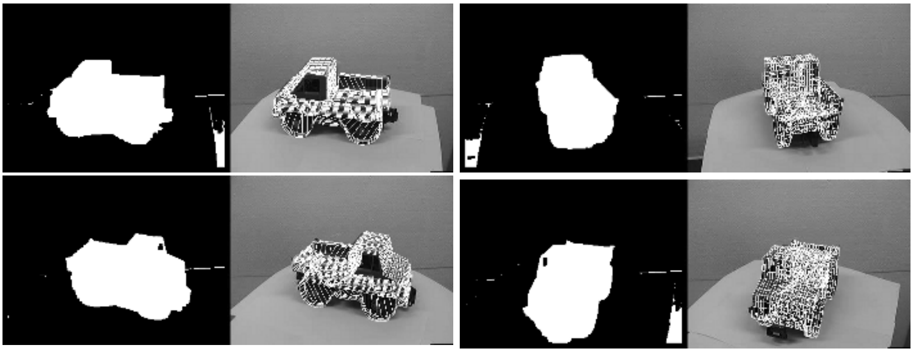


Fig. 10. Pose results of the car model on a turn table and the extracted image silhouettes from which the outline contour is extracted. Note the extraction errors which occur because of shadows and other fragments.

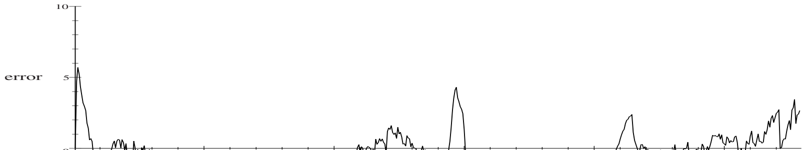


Fig. 11. The absolute error between the estimated angle and the ground truth in degrees. The maximum error is 5.73 degrees and the average error is 1.29 degrees.

error of the image sequence is 1.29 degrees. The errors are mainly dependent on the quality of image feature extraction, the calibration quality and the accuracy of the object model. Note that we are working with a full object model with changing aspects during the 360 degrees rotation.

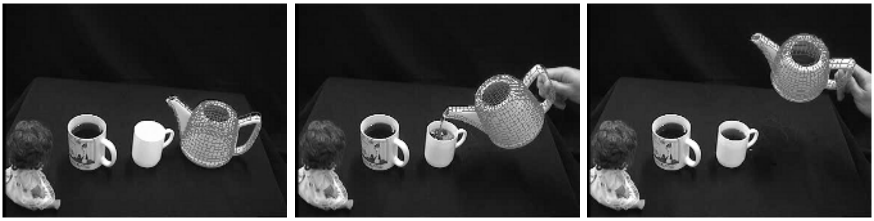


Fig. 12. Example images of the tracked teapot. The hand grasping the teapot leads to outliers during the image silhouette extraction which are detected and eliminated during pose estimation.

4.2 Multiple Component Based Pose Estimation

We now present an extension of our approach for surface based free-form pose estimation to multiple surface patches. The reason is, that several objects can be represented through their including free-form parts more easily. Assume for example a teapot (see e.g. figure 12). It consists of a handle, a container and a spout.

We assume an extracted image silhouette and start with the reconstruction of the image contour points to 3D projection rays. This reconstruction is only estimated once for each image. Then the parts of the object model are projected in a virtual image. Since we assume the surface parts as rigidly coupled we extract and reconstruct **one** 3D silhouette of the surface model. Then we apply the 3D contour on our contour based pose estimation algorithm, which contains an ICP-algorithm and our gradient descent method for pose estimation. We then transform the surface model with the pose calculated from the contour based pose estimation algorithm and increase the low-pass approximation of each surface patch. Since the aspect of the object model can change after the iterated rigid transformations we generate a new 3D silhouette: The algorithm continues with a new projection of the object model in a virtual image and the loop repeats till the algorithm converges.

Figure 12 shows example images during an image sequence containing 350 images. This image sequence shows, that our algorithm is also able to deal with outliers during image processing which are caused by the human hand grasping the teapot.

4.3 Multiple-Components Mixed-Mode Pose Estimation

We now present experimental results where in addition to the silhouette also the internal object information is taken into account as discussed in section 3.3. The effect of using additional internal information is exemplarily shown in figure 13. As can be seen, the opening contour of the teapot is forced to the opening hole in the image and therefore stabilizes the result.

According to our previous experiment of the car on the turn table, we now present a similar experiment with the teapot. Furthermore, we estimate the

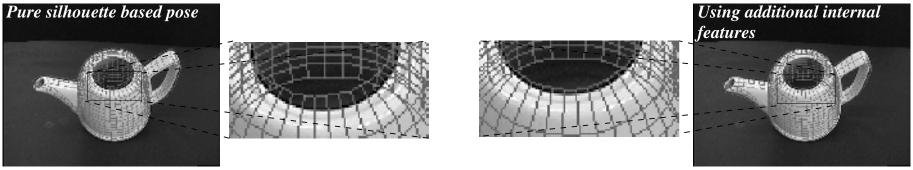

Pure silhouette based pose
Using additional internal features

Fig. 13. Comparison of pose results of the pure silhouette based pose estimation algorithm (left) and the modified one (right) which uses additionally internal edge information.

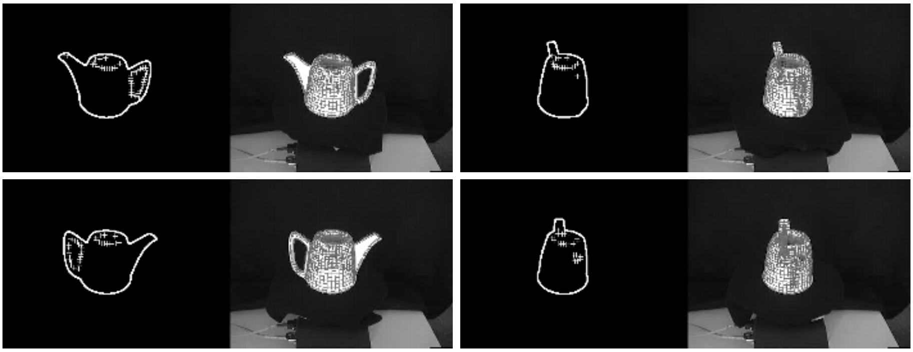


Fig. 14. Example images of the teapot on the turn table. The images to the left show the results after the image processing, the extracted contour and the used internal feature points. The images to the right show pose results.

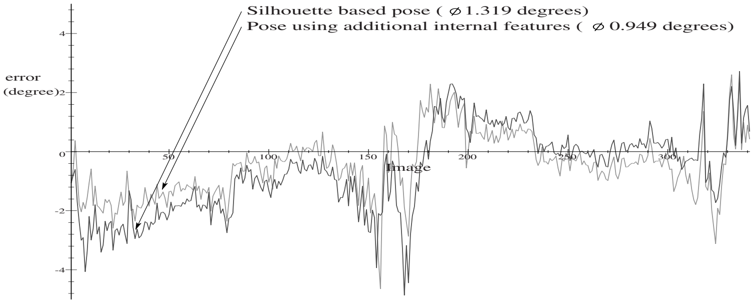


Fig. 15. The absolute pose error of the pure silhouette based pose estimation in comparison to the modified algorithm which uses additional internal object features. The error measure is the angle difference in degrees during the turntable image sequence.

absolute rotational error in degrees between the real pose and the ground truth of the teapot with and without using internal image information. Figure 14 shows example images of the image sequence and the image processing results, the used contour and the used internal image features. Figure 15 shows the comparison of the estimated pose with the angle of the turn table. The average error of the pure silhouette based pose estimation algorithm is 1.32 degrees and the average pose error by using additional internal features is 0.95 degrees. In this sequence we use an image resolution of 384 × 288 pixels and the object is located approximately 1m in front of the camera. The calibration was performed with a calibration pattern containing 16 manually tracked reference points leading to a calibration error of 0.9 pixels for the reference points. This means we only work with coarsely calibrated cameras and low image resolution. The average computing time is 200 ms on a Linux 2GHz machine.

Indeed the comparison holds for just this scenario. For other objects the use of additional internal information might be useless or much more important than the extracted image silhouette. The aim of the experiments is to show that it is possible to extend the silhouette based pose estimation algorithm to scenarios which also use internal edge information of the surface model. To achieve this, we extend the surface model to a combination of free-form surface patches and free-form contour parts.

5 Discussion

In this work we present an extended approach for pose estimation of free-form surface models. Free-form surfaces are modeled by three 2D Fourier descriptors and low-pass information is used for approximation. The estimated 3D silhouette is then combined with the pose estimation constraints. Furthermore, an extension to the use of internal corner features of the object is presented. This leads to a combination of surface models with contour parts which is applied advantageously to the pose estimation problem. To deal with the basic problem of coupling projective geometry and kinematics we use a conformal geometric algebra. Though the equations are only symbolically explained, they present their simple geometric meaning within the chosen algebra. We further present experiments on different image sequences which visualize the properties of our algorithms e.g. in the context of noisy image data. The experiments show the stability of our algorithms with respect to noise and their capacity to deal with aspect changes during image sequences. Since we need up to 200ms per frame, our algorithms are fast and in the area of real-time.

Acknowledgments. This work has been supported by DFG Graduiertenkolleg No. 357, the EC Grant IST-2001-3422 (VISATEC) and by the DFG project 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. Besl P.J. The free-form surface matching problem. *Machine Vision for Three-Dimensional Scenes, Freemann H. (Ed.)*, pp. 25–71, Academic Press, 1990.
3. 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.

4. Campbell R.J. and Flynn P.J. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding (CVIU)*, No. 81, pp. 166–210, 2001.
5. 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.
6. Granlund G. Fourier preprocessing for hand print character recognition. *IEEE Transactions on Computers*, Vol. 21, pp. 195–201, 1972.
7. Grimson W. E. L. Object Recognition by Computer. *The MIT Press, Cambridge, MA*, 1990.
8. Lowe D.G. Solving for the parameters of object models from image descriptions. In *Proc. ARPA Image Understanding Workshop*, pp. 121–127, 1980.
9. Li H., Hestenes D. and Rockwood A. Generalized homogeneous coordinates for computational geometry. In [14], pp. 27–52, 2001.
10. Murray R.M., Li Z. and Sastry S.S. A Mathematical Introduction to Robotic Manipulation. *CRC Press*, 1994.
11. Needham T. Visual Complex Analysis. *Oxford University Press*, 1997
12. 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`
13. 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 Heidelberg, LNCS 2781, pp. 574–581.
14. Sommer G., editor. Geometric Computing with Clifford Algebra. *Springer Verlag*, 2001.
15. 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.
16. Zerroug, M. and Nevatia, R. Pose estimation of multi-part curved objects. *Image Understanding Workshop (IUW)*, pp. 831–835, 1996