Pose estimation based on Geometric Algebra

Yan Cui
DFKI
Trippstadter Strasse. 122
67663 Kaiserslautern Germany
Yan.Cui@dfki.de

Dietmar Hildenbrand
TU Darmstadt
Hochschulstrasse. 10
64283 Darmstadt Germany
Hildenbrand@gris.informatik.tu-darmstadt.de

ABSTRACT

2D-3D pose estimation is an important task for computer vision, ranging from robot navigation to medical intervention. In such applications as robot guidance, the estimation procedure should be fast and automatic, but in industrial metrology applications, the precision is typically a more important factor. In this paper, a new 3D approach for infrared data visualization precisely with the help of 2D-3D pose estimation based on Geometric Algebra is proposed. The approach provides a user friendly interface, a flexible structure and a precise result, which can be adjusted to almost all the geometrically complex objects.

Keywords: Geometric algebra, 2D-3D pose estimation, ICP algorithm.

1 INTRODUCTION

2D-3D pose estimation is an important problem in computer vision. The standard requisites to the pose estimation procedures are high speed, automatic mode and high precision. The main aim in these procedures is to define the relative position and orientation of a known 3D object with respect to a reference camera system. In other words, we search for a transformation (i.e. the pose) of the 3D object such that the transformed object corresponds to 2D image data. For rigid objects, this transformation should be the Euclidean transformation consisting of a rotation R and a translation t. Pose estimation is a subclass of the more general registration problem. The main focus in this paper is given to the pose estimation based on Geometric Algebra and the 3D data visualization with texture mapping. This leads to three main questions:

- How and what kind of image and object features to extract?
- How to do the pose estimation precisely and fast?
- How to detect object parts (surfaces) are visible?

Note that throughout this paper the 3D object model (independent of its representation) is assumed to be known (3D object model is given .wrl file format). The problem how the model of unknown object can be obtained is discussed in works by N. Krueger [21] and M. Zerroug [32].

A 3D object can contain different features like 3D points, 3D lines, 3D spheres, 3D circles, kinematic chain segments, boundary contours and contour parts. The aim is to find the rotation R and the translation t of the object which leads to the best fit of the reference model with the actually extracted entities. So far, it is not defined how to measure the fit quality. It is clear by intuition that a mathematical formalization is not trivial.Current approaches to pose estimation (and registration in general) can be divided into two categories:

- Explicit pose estimation [28]: The involved 2D and 3D entities are defined explicitly. This includes points, lines and higher order entities such as conics, kinematics chains or higher order 3D curves.
- Free-form pose estimation [28]: The involved entities are modeled as free-form objects such as parametric curves/surfaces, 3D meshes, active contours and implicit curves/surfaces.

Additionally, from a statistical point of view, pose estimations of global object descriptions are more accurate and robust than those from a sparse set of local features. But on the other hand, pose estimation based feature can be performed much faster. In this paper we discuss 2D-3D pose estimation using the feature-based method in explicit point corresponding and the free-from method in active contour. After finding the right posed position of the 3D object, we try to visualize the 3D data with texture mapping from the 2D image to the 3D mode, test whether the triangles of the 3D object are visible or not with a ray-tracing algorithm. In this paper we implement the ray-tracing method based on the Geometric Algebra approach in [13].

Main contribution in this work can be generalized as follows:

- We do the camera calibration based on the linear method. This model is used in the geometric algebra framework. The conformal geometric algebra [23] allows to deal with higher order entities (lines, planes, circles, spheres) in the same manner as with points. It is further possible to model the conformal group on these entities by applying special operators in a multiplicative manner.

- This paper introduces a new pose estimation method based on the active object contour extraction. To estimate the pose of free-form contours, ICP (Iterative Closest Point) algorithms [30, 15] are applied. Normal ICP starts with two data sets and an initial guess for their rigid body motion. Then the transformation is refined by repeatedly generating pairs of corresponding point sets and minimizing the error metric. Furthermore, they will later be used to compare a 3D contour, modeled by Fourier descriptors, with 3D reconstructed projection rays. The use of Fourier descriptors is accompanied by some features, which can advantageously be applied within the pose estimation problem: instead of estimating the pose for a whole 3D contour, low-pass descriptions of the contour can be used for an approximation. This leads to a speed up of the algorithm. Meanwhile, this paper brings forward an improved ICP, which improves the normal ICP algorithm to avoid the local minimum.

The paper is structured as follows. In section 2, related work of pose estimation based on the geometric algebra is presented. The 2D-3D entities constraint equations and some experiments of 2D-3D point to line constraints will be given in section 3. Section 4 describes 2D-3D pose estimation based on an active contour method.

2 RELATED WORK

The first pose estimation algorithms were based on a point-based method, which is widely discussed in many foundational papers. A rigid body is generally assumed, but no complete explicit geometric model is given. Methods of this class were firstly studied in the 80's and 90's and pioneering works were done by Lowe [11, 12] and Grimson [22]. Lowe applied a Newton-Raphson minimization method to the pose estimation problem and showed the direct application of numerical optimization techniques in the context of noisy data and in gaining fast (real-time capable) algorithms. Lowe's work is based on pure point concepts and he expresses the constraint equations in the 2D image plane. To linearize the equations, an affine camera model is assumed. The extension to a fully projective formulation is proposed by Araujo et al [1]. The minimum number of correspondences that produce an unique solution are three (non collinear and non-coplanar) points. Four coplanar and non-collinear points also give a unique solution [17]. In general the accuracy increases with the number of used point features. Over-determined solutions are also used for camera calibration [25].

A pose estimation algorithm based on dual quaternions [31] is given by Walker et al. [24]. The method uses the real-part of the dual quaternion to estimate the rotational part and the dual-part of the dual-quaternion to estimate the translational part of the pose. This approach is also discussed by Daniilidis [10] in the context of hand-eye calibration.

There exist some methods that do the pose estimation with image silhouettes (also called occluding contours, extremal contours, apparent contours), which are a rich source of geometric information about the 3D objects. An image silhouette is the projection of the locus of points on the object.

Reconstructing the shape from silhouettes was introduced by Baumgart [4] more than three decades ago. Cippolla and Blake [7] showed that by analysing silhouette deformations local surface curvature can be computed along the corresponding contour generators. Forsyth [8] showed that outlines of algebraic surfaces completely determine their projective geometry from a single view. Cross et al. [9] studied the projective relationship between the coefficients of quadratic algebraic surfaces and the coefficients of the corresponding 2D algebraic silhouettes. Due to perspective projection, the relationship between algebraic surface and algebraic plane curve coefficients is very complex for higher-order surfaces. Kang et al. [18] reconstructed 3D surfaces from occluding contours of algebraic surfaces using a linear dual-surface approach that makes use of the duality between 3D points and tangent planes.

For 2D-3D pose estimation, Kriegman and Ponce [20] parameterised image silhouette equations by 3D pose parameters and minimized the distance between such equations and pixels representing image outlines to obtain the optimal pose. Rosenhahn [28] used the explicit approach instead and back-projected lines through the silhouette pixels in order to register 3D models with those lines. He extended approach to human motion tracking in [29]. Ilic et al. [16] and Knossow et al. [19] also used image silhouettes for human motion tracking using implicit equations.

There are also several variations in the methods of pose estimation. An overview of existing techniques for pose estimation is given by J.S. Goddards PhD-thesis [17].

3 POSE ESTIMATION WITH ENTITIES CORRESPONDENCE

3.1 Pose constraints in conformal geometric algebra

In this section we give a brief framework about how the interaction of entities in geometric algebras are applied on the pose problem. As mentioned earlier, the main problem in the pose estimation is determination of the 2D image features corresponding to 3D object features. The constraint equations can lead to equations of the following equation [28] (this one is just for point correspondences).

$$\lambda\left(\left(M\underline{X}\tilde{M}\right)\underline{\times}e_\infty\wedge(O\wedge x)\right)\cdot e_+ = 0 \quad (1)$$

where λ is a scale parameter, O is the camera position, the underline characters stand for the points in conformal space, the commutator $\underline{\times}$ [26] is used to model a distance measure.

The interpretation of the equation is simple as the equation can be separated in the following manner,

$$\underbrace{\lambda\underbrace{\left(\underbrace{\left(M\underbrace{\underline{X}}_{\text{point in conformal}}\tilde{M}\right)}_{\text{rigid motion}}\underline{\times}e_\infty\wedge\underbrace{\left(\underbrace{O}_{\text{optical center}}\wedge\underbrace{x}_{\text{image point}}\right)}_{\text{projection ray in conformal space}}\right)}_{\text{collinearity of the object point with reconstructed line}}\cdot e_+}_{\text{Euclidean distance measure between line and point}} = 0 \quad (2)$$

We see that the strategy of expressing the pose problem can directly be seen from the equation. All geometric aspects are considered and the equation is compact and easy to interpret.

The main denotes advantages of the constraint equations are:

1. The constraints are expressed in a multiplicative manner, they are concise and easy to interpret. This is the basis for further extensions, like kinematic chains and other higher order algebraic entities.

2. The whole geometry within the scenario is concerned and strictly modeled. This ensures an optimal treating of the geometry and the knowledge that no geometric aspects have been neglected or approximated which is sometimes done in the literature [5] by using orthographic camera models.

3.2 Numerical estimation of pose parameters

In the section 3.1, we give constraint equation that relate 3D object entities to 2D image information. In these equations the object, camera and image information are assumed to be known, the motor M expressing the motion is assumed to be unknown. The main question is now, how to solve a set of constraint equations for multiple features with respect to the unknown motor M. Since a motor is a polynomial of infinite degree, this is a non-trivial task, especially in the case of real-time estimation.

How to get a linear equation with respect to the generators of the motor? We try to solve this problem with exponential representation of motors and the Taylor series expansion with the first approximation order. This leads to a mapping of the above mentioned global motion transformation to a twist representation, which allows for incremental changes of pose. This results in linear equations in the generators of the unknown 3D rigid body motion. In this section the linearization of the motor is derived. For simplicity, we consider the case of point transformations.

The Euclidean transformations of a point X in conformal space caused by the motor M is approximated as:

$$\begin{aligned} M\underline{X}\tilde{M} &= \exp\left(-\frac{\theta}{2}\left(l'+e_\infty m'\right)\right)\underline{X}\exp\left(\frac{\theta}{2}\left(l'+e_\infty m'\right)\right) \\ &\approx \left(1-\frac{\theta}{2}\left(l'+e_\infty m'\right)\right)\underline{X}\left(1+\frac{\theta}{2}\left(l'+e_\infty m'\right)\right) \\ &\approx E+e_\infty\left(x-\theta(l'\cdot x)-\theta m'\right) \end{aligned} \quad (3)$$

We assume $l := \theta l'$ and $m := \theta m'$, then:

$$M\underline{X}\tilde{M} \approx E+e_\infty(x-l\cdot x-m) \quad (4)$$

In the next step we estimate the motion of the 3D object with the previously derived point-line constraint, it leads to

$$\begin{aligned} 0 &= M\underline{X}\tilde{M}\underline{\times}\underline{L} \\ 0 &= \exp\left(-\frac{\theta}{2}\left(l'+e_\infty m'\right)\right)\underline{X}\exp\left(\frac{\theta}{2}\left(l'+e_\infty m'\right)\right)\underline{\times}\underline{L} \\ 0 &\approx \left(E+e_\infty(x-l\cdot x-m)\right)\underline{\times}\underline{L} \\ 0 &= \lambda\left(E+e_\infty(x-l\cdot x-m)\right)\underline{\times}\underline{L} \end{aligned} \quad (5)$$

Due to the approximation $\approx$ in equation (5), the unknown motion parameters l and m are linear. This equation contains six unknown parameters for the rigid body motion. The unknowns are the unknown twist parameters for the motion. In the last step the linearized constraints are scaled with a suitable factor λ to express an Euclidean distance measure as mentioned in section 3.1. This means, all transformations are done in the conformal space, only in the last step the constraint equations are scaled for transformations in the Euclidean space.

The linear equations are solved for a set of correspondences by applying the Householder method [27]. From the solution of the system of equations, the motion parameters R, t can easily be recovered by evaluating $\theta := \|l\|, l' := \frac{l}{\theta}, m' := \frac{m}{\theta}$. The motor M can be evaluated by applying the Rodrigues' formula.

The principle of this approximation is illustrated in figure 1. The aim is to rotate a point $\underline{X}$ by 90 degrees to a point $\underline{X}'$. The first order approximation of the rotation leads to the tangent of the circle passing through X.

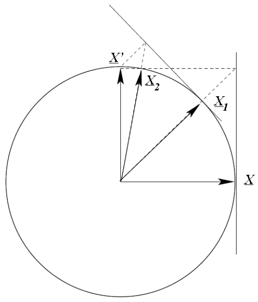

Figure 1: Principle of convergence for the iteration of a point X rotated around 90 degrees to a point X'. X_1 is the result of the first iteration and X_2 is the result of the second iteration. [28]

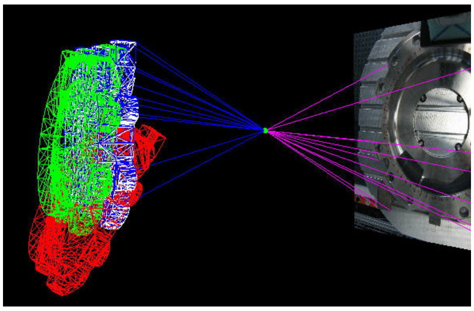

Figure 2: Iterative pose estimation process, the red object is the position after the first iteration, the green one is after the second iteration, the blue and the white objects are the positions after the third and fourth iteration.

Normalizing the tangent line to $\underline{X}'$(denoted by dashed lines) $\underline{X}_1$ is gained as the first order approximation of the required point $\underline{X}'$. By repeating this procedure the points $\underline{X}_2 \ldots \underline{X}_n$ will be estimated, approaching to the point $\underline{X}'$. It is clear from figure 1 that the convergence rate of a rotation depends on the amount of the expected rotation.

All angles converge during the iteration. For the most cases just a few iterations are sufficient to get a good approximation. In situations where only small rotations are assumed, four iterations are sufficient for all cases.

3.3 Result

We use the point-line constraint to construct the linear equation matrix and the least square method to solve it. Four iterations are needed to compute the final translation t and rotation R. The developed algorithm interactively computes the camera position (see Figure 2).

The next problem that we need to solve for texture mapping is detection of visible areas. We should detect visible triangle for the computed camera position. To solve the problem, the Ray-tracing algorithm is used: For each point, there exists a ray from this point to the camera position. If the ray hits a triangle of the object before it gets to the camera position, we consider the point is invisible, otherwise it is visible. If three points of the triangle are visible, we consider this triangle as visible. Ray-tracing algorithm gives the texture coordinates in the 2D image, which are used in texture mapping. There're some experiments results in Figure 3, on the left are the original image, the green points are the corresponding points between 2D image and 3D object detected by user, on the right are the texuted object. We see clearly that the developed algorithm successfully solve the texture mapping problem.

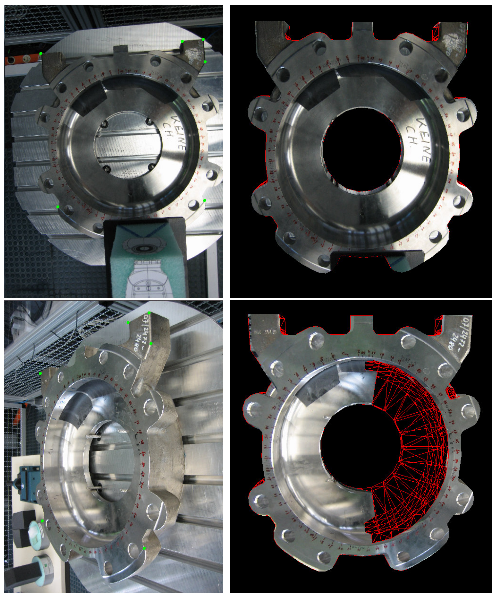

Figure 3: Left: 2D image, the green points are the corresponding points. Right: final textured 3D object.

Now we present a practical application of the developed algorithm for non-destructive testing (NDT). Thermal inspection is one of the numerous methods in NDT. The inspection consists of two cases: 1) excitation using the flash lamps 2) observation of the cooling process using an infrared camera. The existing methods in representation of infrared data involve 1D (time profile) and 2D (x/y space) forms. They're caused by using focal plane array (FPA) detectors in infrared cameras. The technique developed in this paper significantly extends capabilities in representation of acquired data. The combination with prior known geometry of an object to be inspected makes the representation more informative and allows analyzing the physical processes inside the object taking into account its geometry. The developed algorithm was successfully tested for visualization of thermal inspection data, Figure 4 shows the infrared 3D data sequence visualization as time increase.

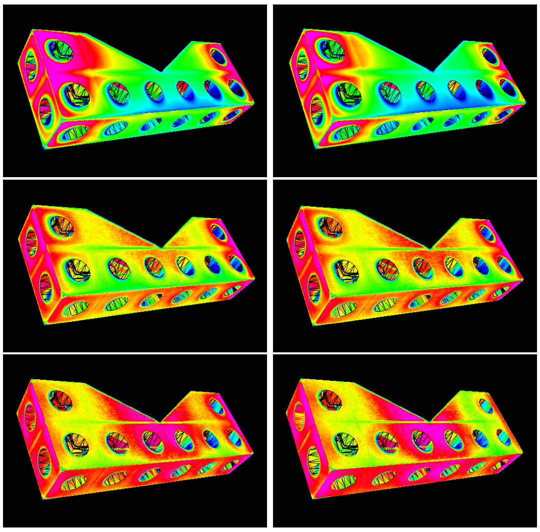

Figure 4: From Left to right, from top to bottom: 3D Visualization of the thermal image set.

4 POSE ESTIMATION WITH ACTIVE CONTOUR CORRESPONDENCE

4.1 3D object contour in Fourier domain

This section we intorduce signal theoretic foundations. The aim is to define the discrete Fourier transformation and its extension to the 3D space in classical matrix calculus. More detail information can be found in [3, 2].

For 3D contour interpolation a set $f_j^3 \in \Re^3$ of 3D values is assumed $j = 0, \ldots, M-1, M \in \aleph$. These values are contour points of a closed contour. To achieve a 3D contour interpolation, the 3D signal can be interpreted as 3 separate 1D signals:

$$F_m^3 = \frac{1}{M} \sum_{u=0}^{M-1} \begin{pmatrix} f_u^3(1) \\ f_u^3(2) \\ f_u^3(3) \end{pmatrix} \exp\left(\frac{-2\pi ium}{M}\right) \quad (6)$$

And its inverse transformation can be written as

$$f_u^3 = \sum_{m=0}^{M-1} \begin{pmatrix} F_m^3(1) \\ F_m^3(2) \\ F_m^3(3) \end{pmatrix} \exp\left(\frac{2\pi imu}{M}\right) \quad (7)$$

Taking only a subset of the phase vectors leads to a low-pass approximation of the contour. This is applied to speed up the algorithm for pose estimation of free-form contours and to avoid local minimum during iterations.

The user should give initial position of the 3D object firstly, and then select the region or sub-region of the 3D object which is captured in the 2D image. Finally the selected object will be mapped to 2D image, as Figure 5 left shows. This 2D information can help us to find the discrete contour points of the 3D object. The contour points of the mapped 2D image correspond to the discrete contour points of the 3D object. As Figure 5 middle shows, the algorithm can find the discrete contour points. With Fourier transformation as talked above, we can get the continuous contour of the 3D object from the discrete contour points (as Figure 5 right shows). The user can also select the sub-region of the 3D object and with the same processing, get the continuous contour of the 3D object.

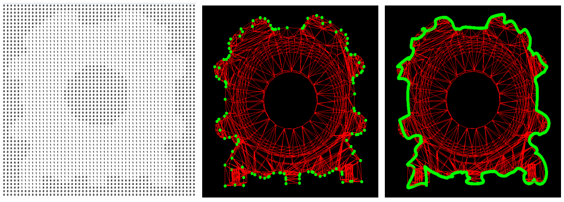

Figure 5: Left: Mapped 2D image from 3D object. '1' is the object region, '0' is the background region. Middle: Discrete contour points of the 3D object. Right: Continuous contour of the 3D object.

4.2 2D image contour

2D active image extraction algrithom [6] is proposed an active contour model based on Mumford-Shah segmentation technique and the level set method. The model is not based on an edge-function to stop the evolving curve on the desired boundary. Also, we do not need to smooth the initial image, even if it is very noisy and in this way, the locations of boundaries are very well detected and preserved. By this model, we can detect objects whose boundaries are not necessarily defined by gradient or with very smooth boundaries, for which the classical active contour models are not applicable. The position of the initial curve can be anywhere in the image, and it does not necessarily surround the objects to be detected.

Let us define the evolving curve C in Ω, as the boundary of an open subset ω (i.e. $\omega \subset \Omega$, and $C = \partial\omega$). Then, $inside(C)$ denotes the region ω, and $outside(C)$ denotes the region $\Omega\backslash\omega$. This method is the minimization of an energy based-segmentation. Let us first explain the basic idea of the model in a simple case. Assume that the image u_0 is formed by two regions of approximatively piecewise-constant intensities, of distinct values u_0^i and u_0^o. Assume further that the object to be detected is represented by the region with the value u_0^i. We denote its boundary initially by C_0. Then we have $u_0 \approx u_0^i$ inside the object (or $inside(C_0)$), and $u_0 \approx u_0^o$ outside the object (or $outside(C_0)$). Now let us consider the following 'fitting' energy function:

$$\begin{aligned} F(c_1, c_2, C) = \quad & \mu L(C) + \nu A(in(C)) \\ & + \lambda_1 \int_{in(C)} |u_0(x,y) - c_1|^2 dxdy \\ & + \lambda_2 \int_{out(C)} |u_0(x,y) - c_2|^2 dxdy \end{aligned} \quad (8)$$

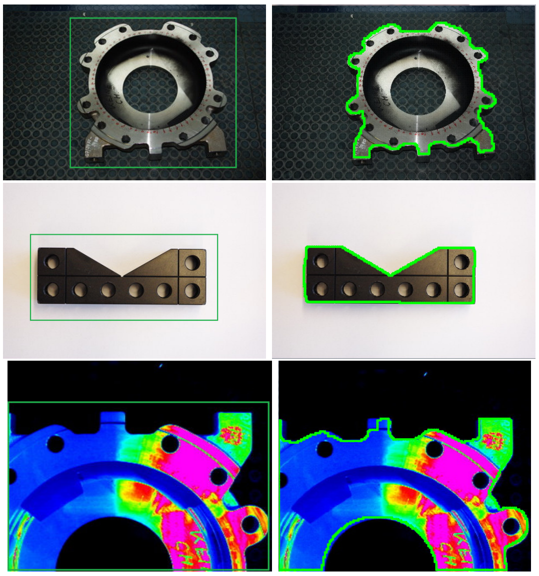

Figure 6: Left: 2D image and the initial contour given by the user (green line). Right: 2D image contour given by the active contour algorithm (green line)

Where $L(C)$ stands for the length of the contour, $A(in(C))$ stands for the area in the contour, c_1 and c_2 is are the average intensity levels inside and outside of the contour. $\mu, \nu, \lambda_1, \lambda_2$ are the weight parameters. There're some algorithms to find the minimization of the energy function, Therefore, we consider the minimization problem:

$$\inf_{c_1,c_2,C} F(c_1,c_2,C) \tag{9}$$

Simply we can consider the Euler-Lagrange equation to solve this problem. With this method the contour of the object can be extracted reiteratively. THe final results are shown in figure 6, on the left are the original images, the green lines are the initial contour defined by user, on the right are the contour results with the green line showed.

4.3 Pose estimation between 2D image and 3D object's contour

The aim is to formulate a 2D-3D pose estimation algorithm for any kind of free-form contour. The assumptions are the following:

- The object contour curve is given as a set of 3D points f_j^3, spanning the 3D contour.
- In an image of a calibrated camera, the object is observed in the image plane and a set of 2D points x_j^2 spanning the 2D contour is extracted.

Since the number of contour points in the image is often too high (e.g. 800 points in the experimental scenario), just every kth point (e.g. $k \in 5,\ldots,20$) is used to get an equal sub-sampled set of contour image points.

Note that there is no knowledge which 2D image point corresponds to the 3D point of the interpolated 3D model contour. Furthermore, a direct correspondence does not generally exist since the contours are mostly sampled from different starting points and the number of image and object points may also vary.

Using the approach for pose estimation of point-line correspondences, the Iterative Closest Point (ICP) [30] algorithm for free-form contours consists of iterating the following steps:

Algorithm 1: Normal ICP Algorithm

1. Reconstruct projection rays from the image points.
2. Estimate the nearest point of each projection ray to a point on the 3D contour.
3. Estimate the pose of the contour with the use of this correspondence set.
4. Goto 2.

The idea is that all image contour points simultaneously pull on the 3D contour. This is the normal ICP algorithm, and there exist two aspects to improve the performance.

- With Fourier transformation, increasing degree method can improve the calculation speed.
- We can improve normal ICP to avoid the local minimum problem.

We talk about the methods in detail. Increasing degree method: Using the Fourier coefficients for contour interpolation works well, but the algorithm can be made faster by using a low-pass approximation for pose estimation and by adding successively higher frequencies during the iteration. We call this technique the increasing degree method. Therefore the pose estimation procedure starts with just a few Fourier coefficients of the 3D contour and estimates the pose to a certain degree of accuracy. Then the order of used Fourier coefficients is increased and the algorithm proceeds to estimate the pose with the refined object description. Improve ICP to avoid the local minimum: We can define the error, which is sum of the distances between the posed object points and the nearest rays from the image points. The user can define a threshold, for our experiments, we define the threshold 0.005. If the error is bigger than the threshold, the ICP comes to the local minimum. Then rotate the image space, such as 10 degree around the view angle, then do the normal ICP again, do this procedure again and again, until the ICP get the error smaller than the threshold. The algorithm pipeline is as follows:

With the improved ICP algorithm, the performance of pose estimation results are presented in Figure 7.

Algorithm 2: Improved ICP Algorithm

1. If error > threshold (0.005), Rotation of the image. (10 degree around the view angle).

2. Pose estimation, do this step 4 times

   (a) Reconstruct projection rays from the image points.

   (b) Estimate the nearest point of each projection ray to a point on the 3D contour points, which is produced by the Fourier interpolation.

   (c) Estimate the pose of the contour with the use of this correspondence set.

   (d) Increasing the Fourier coefficients of the 3D object contour, goto (b).

3. Calculation of the new error. Goto 1.

End

5 CONCLUSION

The main focus concentrates on pose estimation based on Geometric Algebra and 3D data visulazation with texture mapping. 3D object models are treated feature based and active contour form based: The results of this paper are summarized in the following points:

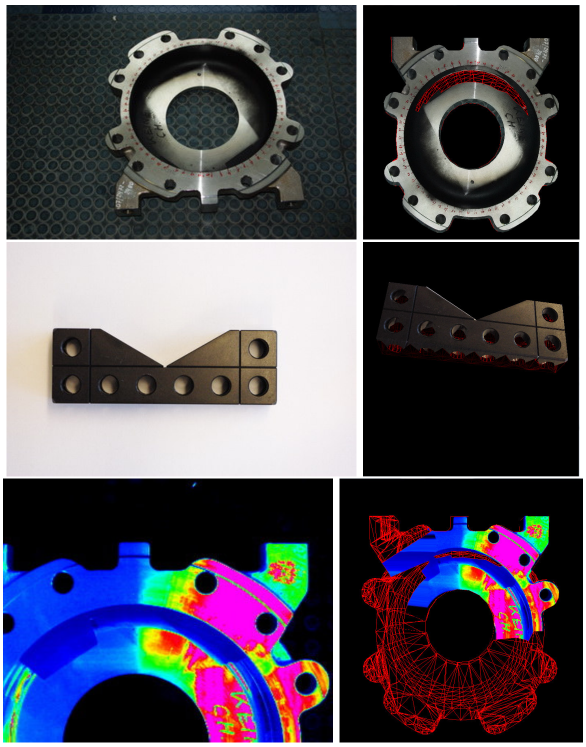

Figure 7: Left: 2D image. Right: textured 3D object after contour corresponding estimation.

- The geometry of the 2D-3D pose estimation scenario is analyzed and the interaction of entities given in conformal space. It leads to a compact and linear description of the pose problem which contains a distance measure. These equations can further be scaled by a scalar which allows for an adaptive weighting of the constraints. The constraint equations are solved by linearizing and iterating the equations. The estimation of pose parameters is high performance.

- The approach for modeling curves is related to model 3D contours by using Fourier descriptors. In this context ICP algorithms are used to estimate the correspondences and poses for image contours and object contours. The use of low-pass information enables one further to avoid local minimum and to speed up the algorithm. Furthermore, an automatic method avoid the local minimum is possible, which stabilizes the pose results.

The next extension of contour based free-form pose estimation is pose estimation of free-form surfaces. This has a much higher degree of complexity, similar to the extension of the 1D analytic signal and 1D quadrature filters to 2D in an isotropic way, as presented in [14].

Though the ICP algorithm works fine and stable in tracking situations, its computational overhead leads to hardly realizable real-time systems for complex object models. Especially for the camera calibration, it's difficult to realizable a stable and real-time performance. Here also some work is possible and promising. E.g. no fast Fourier transformation is applied so far and the minima-search in the gradient descent method is highly parallelizable. But maybe new search strategies are better suited than the used ICP algorithm.

Another extendable topic is the image processing for pose estimation. So far easy scenarios are assumed, e.g. with little background noise. The image processing is kept simple to extract the image contour, since the geometric aspects of the pose scenario are dealt with in this paper.

This leads to further extensions for computer graphics or geometric algebra and is an interesting topic for future research.

REFERENCES

[1] H. A. aposujo, R. L. Carceroni, and C. M. Brown. A fully projective formulation for lowe aposujo tracking algorithm. Technical report, 1996.

[2] J. B. Digitale bildverarbeitung. *Springer-Verlag, Berling, Heidelberg, New-York*, 1997.

[3] K. B. Fouriertransformation fuer ingenieur- und naturwissenschafte. *Springer-Verlag, Berlin, Heidelberg, New York*, 2001.

[4] B. Baumgart. *Geometric Modeling for Computer Vision*. PhD thesis, stanford University, 1974.

[5] C. Bregler and J. Malik. Tracking people with twists and exponential maps, 1998.

[6] T. F. Chan and L. A. Vese. Active contours without edges, 2001.

[7] R. Cipolla, G. Fletcher, and P. Giblin. Surface geometry from cusps of apparent contours. In *In: ICCV*, pages 858–863, 1995.

[8] F. Computer and D. A. Forsyth. Recognizing algebraic surfaces from their outlines. In *In International Conference on Computer Vision*, pages 476–480, 1992.

[9] G. Cross and A. Zisserman. Quadric surface reconstruction from dual-space geometry. *In Proceedings of the 6th International Conference on Computer Vision,Bombay, India*, pages 25–31, January 1998.

[10] K. Daniilidis. Hand-eye calibration using dual quaternions. *International Journal of Robotics Research*, 18:286–298, 1998.

[11] L. D.G. Solving for the parameters of object models from image descriptions. *in Proc. ARPA Image Understanding Workshop*, pages 121–127, 1980.

[12] L. D.G. Three-dimensional object recognition from single twodimensional images. *Artificial Intelligence*, Vol. 31, No. 3:355–395, 1987.

[13] L. Dorst. *Geometric Algebra for Computer Science An Object -Oriented Approach to Geometry, Chapter 23.* 2008.

[14] FelsbergM. Low-level image processing with the structure multivector. Technical report, Technical Report 0203,Christian-Albrechts-Universitaet zu Kiel, Institut fuer Informatik und Praktische Mathematik, 2002.

[15] D. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets, 2001.

[16] S. Ilic, M. Salzmann, and P. Fua. Implicit surfaces make for better silhouettes,cvpr 05, 1135.

[17] G. J.S. *Pose and Motion Estimation From Vision Using Dual Quaternion-Based Extended Kalman Filtering.* PhD thesis, Springer-Verlag, New York Inc, 2001.

[18] K. Kang, J.-P. Tarel, R. Fishman, and D. Coope. A linear dual-space approach to 3d surface reconstruction from occluding contours using algebraic surfaces. In *In International Conference on Computer Vision*, pages 198–204, 2001.

[19] D. Knossow, R. Ronfard, R. Horaud, and F. Devernay. F.: Tracking with the kinematics of extremal contours. In *Computer Vision ¨C ACCV 2006. LNCS*, pages 664–673. Springer, 2006.

[20] D. J. Kriegman and J. Ponce. On recognizing and positioning curved 3-d objects from image contours. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 12:1127–1137, Dec 1990.

[21] N. Krueger, M. Ackermann, G. Sommer, and L. F. K. Systeme. Accumulation of object representations utilizing interaction of robot action and perception. *Knowledge Based Systems*, 13:200–2, 2002.

[22] G. W. E. L. *Object Recognition by Computer*. The MIT Press,Cambridge, MA, 1990.

[23] H. D. Li H. and R. A. A. generalized homogeneous coordinates for computational geometry. pages 27–52, 2001.

[24] W. M.W. and S. L. Estimating 3-d location parameters using dual number quaternions. *Computer Vision, Graphics, and Image Processing (CVGIP): Image Understanding*, Vol. 54, No. 3:358¨C367, 1991.

[25] F. O. *Three-Dimensional Computer Vision, A Geometric Viewpoint.* MIT Press, Cambridge, 1993.

[26] C. Perwass. *Applications of Geometric Algebra in Computer Vision*. PhD thesis, Cambridge University, 2000.

[27] V. W. Press W.H., Teukolsky S.A. and F. B.P. *Numerical Recipes*. Cambridge University Press, 1994.

[28] B. Rosenhahn. *Pose Estimation Revisited*. PhD thesis, Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universit"at zu Kiel, 2003.

[29] B. Rosenhahn, U. G. Kersting, A. W. Smith, J. K. Gurney, T. Brox, and R. Klette. A system for marker-less human motion estimation. In *Page 6 [22] Rosenhahn B., Klette*, pages 45–51. Springer, 2005.

[30] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *INTERNATIONAL CONFERENCE ON 3-D DIGITAL IMAGING AND MODELING*, 2001.

[31] B. W. *Kinematik und Quaternionen, Mathematische Monographien 4.* Deutscher Verlag der Wissenschaften, 1960.

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