STRUCTURAL ICP ALGORITHM FOR POSE ESTIMATION BASED ON LOCAL FEATURES

Marco A. Chavarria and Gerald Sommer

Cognitive Systems Group. Christian-Albrechts-University of Kiel, D-24098 Kiel, Germany

Keywords: Pose estimation, ICP algorithm, monogenic signal.

Abstract: In this paper we present a new variant of the ICP (iterative closest point) algorithm for finding correspondences between image and model points. This new variant uses structural information from the model points and contour segments detected in images to find better conditioned correspondence sets and to use them to compute the 3D pose. A local representation of 3D free-form contours is used to get the structural information in 3D space and in the image plane. Furthermore, the local structure of free-form contours is combined with orientation and phase as local features obtained from the monogenic signal. With this combination, we achieve a more robust correspondence search. Our approach was tested on synthetical and real data to compare the convergence and performance of our approach against the classical ICP approach.

1 INTRODUCTION

Many actual applications in robotics and computer vision deal with objects modelled by e.g. 3D free-form contours. Such models are widely used for problems like monocular and binocular pose estimation and object recognition among others. The more information available about the nature of these entities, the better are the chances to solve the correspondence problem in a more efficient and robust way. With respect to contour models, the simplest and most common representation in the literature uses parametric functions (Zhang, 1994). Active contour models, also known as "snakes" are also widely used for motion tracking and stereo matching (Kass et al., 1987).

Recently, geometric algebra (Sommer, 2001) has been introduced in computer vision as a problem adaptive algebraic language in case of modelling geometric related problems. It turned out that the conformal geometric algebra (CGA) is especially useful because its ability of handling stratified geometrical spaces (Rosenhahn and Sommer, 2005a). The basic geometrical entities (e.g. points, lines and planes) can be embedded in the conformal space, see (Rosenhahn and Sommer, 2005a). Also the rigid body motion has a linear representation (called motor) with respect to

all geometric entities derived from spheres. In the work (Rosenhahn et al., 2004), sets of coupled twists are used to model free-form contours and surfaces in the framework of conformal geometric algebras. In a further work (Rosenhahn and Sommer, 2005b) the pose estimation constraints (point-line, point-plane and line-plane) were also used in that algebra. We propose a new local representation of free-form contours which allows to extract local structural information, which can be also embedded in CGA. Thus, it is also compatible with the pose estimation constraints.

Finding correspondences is one of the most challenging problems for computer vision applications. Two points correspond to each other if a similarity criteria is fulfilled. The most common and simple approach is the ICP algorithm (Besl and McKay, 1992). Zhang (Zhang, 1994) uses a modified ICP algorithm to deal with the occlusion problem. ICP algorithms combined with different metrics are also used, for example point-point (Benjemaa and Schmitt, 1997) and point-line (Dorai et al., 1997). Chen and Medioni (Chen and Medioni, 1992) use the sum of square distance between scene and model point in their ICP variant. An extension of this work was made by Dorai and Jain (Dorai et al., 1997), where an optimal uniform weighting of points is used. A compari-

son of variants of the ICP algorithm is presented in (Rusinkiewicz and Levoy, 2001), where the different variants are applied to align artificially generated 3D meshes. The above cited methods assume that the scene is almost aligned with the model (tracking assumption). Since these variants use as feature only point information, it is possible to optimize the algorithms for real time applications. Other methods combine the ICP algorithm with other image processing approaches like optical flow (Rosenhahn et al., 2006) or bounded Hough transform (Shang et al., 2005). These methods seem to be robust but they are very time consuming, not suitable for real time applications. All the methods based on punctual information have to consider the tracking assumption in order to perform efficiently. For the case of the ICP variants combined with complex image processing approaches, the tracking assumption can be slightly overcome in some cases.

The basic variant of the ICP algorithm finds corresponding point pairs (image-model) by measuring the minimal Euclidean distance. In this case point coordinates can be considered as a local feature. One important question when analyzing local features is how "local" actually the feature should be. The minimal entity which can be described is a point. The only feature available is its position in the 3D space. A single point does not give much information about the object in general. From two neighbor points the local orientation can be derived and three neighbor points are enough to get local curvature. As the neighborhood is increased, more feature information can be extracted and therefore, more information about the nature of the object.

In this paper we present a new variant of structural ICP algorithm, which integrates local features (from model and image) and the structural phase information delivered from the monogenic signal (Felsberg and Sommer, 2001). One advantage of our ICP variant is that it can be perfectly applied for free-form contours and it is robust against the tracking assumption. A local 3D contour representation is used to extract a feature set for contour segments, like concavity, convexity and straightness. Our ICP variant reaches a compromise between computational cost and robustness against the tracking assumption.

For image feature extraction we use the monogenic scale-space approach presented by Felsberg and Sommer (Felsberg and Sommer, 2004), which is briefly described in section 2. In section 3 we introduce the local representation of 3D contours based on local motors. The feature set is obtained from a single motor and the extended set is obtained from contour segments. The ICP structural algorithm is introduced in Section 4. Finally, in Section 5 experiments made on synthetical and real data are presented to validate the efficiency and robustness of our algorithm.

2 IMAGE FEATURES IN SCALE-SPACE

The monogenic scale-space representation and phase-based image processing techniques were introduced in (Felsberg and Sommer, 2004). If $p(\mathbf{x};s)$ and $q(\mathbf{x};s)$ are the filter responses of an image convolved with the Poisson and conjugate Poisson kernels respectively, local amplitude $a(\mathbf{x};s)$ and phase $\mathbf{r}(\mathbf{x};s)$ are obtained for a scale s as shown in equation (1).

$$
\begin{aligned}
a(\mathbf{x};s) &= \sqrt{|q(\mathbf{x};s)|^2 + |p(\mathbf{x};s)|^2} \\
\mathbf{r}(\mathbf{x};s) &= \frac{q(\mathbf{x};s)}{|q(\mathbf{x};s)|}\arctan\left(\frac{|q(\mathbf{x};s)|}{p(\mathbf{x};s)}\right).
\end{aligned}
\tag{1}
$$

The local amplitude is related to the local energy of the signal. The local orientation and local phase are combined in the local phase vector. The local phase gives information about the local symmetry of the signal and the local orientation gives the orientation of the highest signal variance. Then, for an edge point we chose the local features orientation and phase angles in x and y directions $F_i^{im} = \{\phi_i, \| r_i^x \|, \| r_i^y \|\}$.

Once that the local amplitude and phase are obtained for a scale factor s, a contour search algorithm based on the local amplitude and orientation is applied to extract the contour segments. By changing the scale factor, low contrast edges can also be detected.

3 LOCAL CONTOUR REPRESENTATION

The idea of the local representation is to construct a motor to approximate a contour segment. A motor is parameterized by a rotation axis and angle. This is illustrated in the figure 1. A plane is constructed with the 3D points $\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}$, which is parameterized by its normal $\mathbf{n}$ and distance to the origin d. In that plane, a local coordinate system is defined by

$$
\mathbf{i}_1 = \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|}, \quad \mathbf{i}_2 = \mathbf{n}, \quad \mathbf{i}_3 = \frac{\mathbf{i}_1 \times \mathbf{i}_2}{\|\mathbf{i}_1 \times \mathbf{i}_2\|}.
\tag{2}
$$

To find the rotation axis of the motor we need to calculate the center of the circle. To make the computations easier, the problem is translated from 3D to 2D. That is the plane defined by the basis vectors $\mathbf{i}_1$ and $\mathbf{i}_3$ (see right picture of figure 1). The center of the
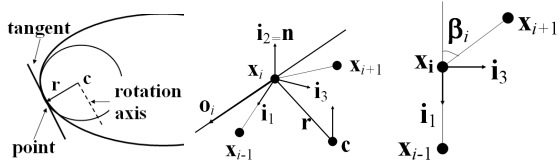
Figure 1: Local motor for a 3D contour (left). Local coordinate system (middle) needed to get the circle parameters of the motor and the structural features (right).

circle $\mathbf{c}$ and the radius vector $\mathbf{r}$ are easily calculated in 2D. Then the coordinates of the center of the circle in 3D are recovered. Thus, the rotation axis of the motor in 3D is obtained with the center $\mathbf{c}$ and the normal vector $\mathbf{n}$. The rotation angle θ_i is the angle defined by the segment $\overline{\mathbf{x}_{i-1}\mathbf{c}\mathbf{x}_{i+1}}$. Finally the orientation vector $\mathbf{o}_i$ is defined by the orthogonal to the radius vector $\mathbf{r}$

For every point of the 3D contour the local curvature vector and bending angle are calculated by

$$\begin{aligned}\mathbf{k}_i &= (\mathbf{x}_i - \mathbf{x}_{i-1}) \times (\mathbf{x}_{i+1} - \mathbf{x}_i) \\ \beta_i &= \mathrm{acos}\frac{(\mathbf{x}_i-\mathbf{x}_{i-1})\cdot(\mathbf{x}_{i+1}-\mathbf{x}_i)}{\|(\mathbf{x}_i-\mathbf{x}_{i-1})\|\|(\mathbf{x}_{i+1}-\mathbf{x}_i)\|}\end{aligned} \quad, \quad (3)$$

where the points $\mathbf{x}_i$, $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i-1}$ are considered in the local coordinate system. In this case, the $\mathbf{e}_3$ component of the resulting curvature vector $\mathbf{k}_i = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + x_3\mathbf{e}_3$ changes its sign when the point is concave or convex. When the scalar x_3 has a negative sign, the point is considered locally convex. Otherwise, it will be locally concave. If the bending angle β_i has a value closed to zero, the point is considered as a part of a straight line.

An extended feature set allows to get more robust features, especially in the image plane where noise is present and digital contours are extracted. In this case we are getting features not only from a single point. The neighborhood of the point is extended to larger segments in order to take average feature values as shown in equation (4).

$$\begin{aligned}\mathbf{k}_i &= \frac{1}{m}\sum_{j=1}^m \mathbf{v}_1 \times \mathbf{v}_2 \\ \beta_i &= \frac{1}{m}\sum_{j=1}^m \mathrm{acos}\frac{\mathbf{v}_1\cdot\mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|}\end{aligned} \quad, \quad (4)$$

where $\mathbf{v}_1 = \mathbf{x}_i - \mathbf{x}_{i-j}$ and $\mathbf{v}_2 = \mathbf{x}_{i+j} - \mathbf{x}_i$.

By taking the point $\mathbf{x}_i$ as a reference, motors are constructed iteratively with the adjacent points. Then the contour segment is defined by the points $\{\mathbf{x}_{i-j}\cdots\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}\cdots\mathbf{x}_{i+j}\}$ and the features of that point corresponds to the structure of the neighborhood.

3.1   3d and 2d Contour Features

We define the following structural features for a 3D point $\mathbf{x}_i$ by

$$F_i^{3D} = \{\mathbf{o}_i, \mathbf{k}_i, \beta_i\}, \quad (5)$$

where $\mathbf{o}_i$ is the local orientation vector at the point $\mathbf{x}_i$, $\mathbf{k}_i$ is the curvature vector and β_i the bending angle. To get the corresponding 2D features, the contour model points are projected onto the image plane (see figure 2), motors are constructed and the features are calculated as described in the last section with the corresponding points in image coordinates $\mathbf{x}'_{i-1}$, $\mathbf{x}'_i$ and $\mathbf{x}'_{i+1}$. The normalized orientation vector $\mathbf{o}'_i$ is obtained and its corresponding orientation angle α_i.

The concept of phase in the image plane delivers information of the local structure of the image derived from the monogenic signal. In the case of edges, the phase encodes a transition from one gray value to another in x and y directions. For 3D contours it is not possible to compute directly phase information in that sense. Despite of that, it is possible to assign a feature value for a projected 3D contour point that represents such transition. We call this feature transition index. Figure 2 shows the idea of transitions t_x and t_y for a point. The transition takes the values $+1$ or -1 (equivalent to the phase responses $\| r_i^x \|$ and $\| r_i^y \|$) depending on the orientation of the vector $\mathbf{o}'_i$. Thus, for a projected 3D contour point we obtain as features the orientation and transition indexes in x and y directions $F_i^{con} = \{\alpha_i, t_i^x, t_i^y\}$.
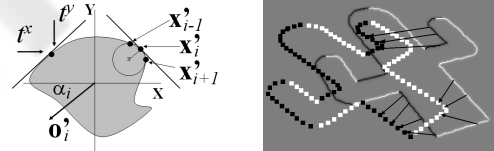

Figure 2: Example of motor construction and the transition index in the image plane (left). Transition index of an projected model contour and phase response of the monogenic signal (right). The lines show the corresponding pairs of image and model.

4   STRUCTURAL ICP VARIANT

Our ICP variant combines error metrics with image feature constraints. Thus, in the image plane we have the following feature sets for projected model segments $F_i^{2Dm} = \{\alpha_i, t_i^x, t_i^y, \mathbf{k}_i^{2Dm}, \beta_i^{2Dm}\}$ and for detected contour segments $F_i^{2Dp} = \{\phi_i, \| r_i^x \|, \| r_i^y \| , \mathbf{k}_i^{2Dp}, \beta_i^{2Dp}\}$. Two points (image and model) form a correspondence pair if the structural constraints are

met. The phase-transition index constraint is defined as

$$C_1 = \left\{ \begin{array}{ll} 1 & \text{if } \| r_i^x \| = t_i^x \wedge \| r_i^y \| = t_i^y \\ 0 & \text{otherwise} \end{array} \right. \quad (6)$$

In the following we will use the sign $\wedge$ to denote the logical "and" operation. The straightness constraint is defined from the local bending angles β_i^{2Dm} and β_i^{2Dp} as

$$C_2 = \left\{ \begin{array}{ll} 1 & \text{if } \beta_i^{2Dm} < t \wedge \beta_i^{2Dp} < t \\ 0 & \text{otherwise} \end{array} \right. , \quad (7)$$

where t is a threshold value. Finally, the concavity-convexity constraint is defined from the sign of the e_3 component of the vectors $\mathbf{k}_i^{2Dm} = x_1 \mathbf{e}_1 + x_2 \mathbf{e}_2 + x_3 \mathbf{e}_3$ and $\mathbf{k}_i^{2Dp} = y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2 + y_3 \mathbf{e}_3$ by

$$C_3 = \left\{ \begin{array}{ll} 1 & \text{if } \text{sign}(x_3) = \text{sign}(y_3) \wedge C_2 = 0 \\ 0 & \text{otherwise} \end{array} \right. \quad (8)$$
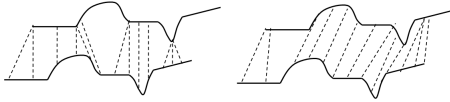

Figure 3: Example of correspondence pairs for normal (left) and structural (right) ICP variants.

The figure 3 shows the idea of ICP combined with structural constraints (straight, concave or convex). The figure on the left shows the case where only the minimal distance is considered, on the right for the structural variant. As can be seen, for a point in the bottom curve, its corresponding point in the upper curve will be the nearest point with the same local structure. This is analogous for the ICP plus the phase-transition index constraint, see left picture of figure 2.

5 EXPERIMENTS

We used for our experiments 3D planar contour models (see figure 4) rich in structure like the "cactus" and "puzzle" models and also the "mouse" model, which has less structure. In the first experiment we compare the convergence behavior of a normal ICP algorithm and our structural ICP variant. The initial position of the model is known, then it is translated and rotated to its actual position and projected onto the image plane to generate an artificial image. On this artificial image the corresponding contour segments and the local features are extracted. Then the pose is calculated and compared with the ground truth. For these experiments relatively large displacements were applied to
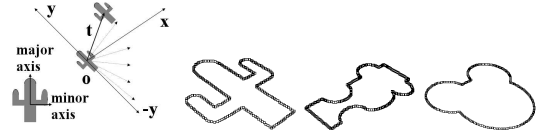

Figure 4: The object is translated in all directions in the plane. For every translation the pose is calculated and compared with the ground truth (left). Different models used in the experiments (right): cactus, puzzle and mouse models.
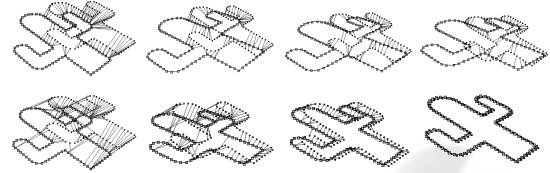

Figure 5: Convergence sequence for normal (top row) and structural (bottom row) ICP variants applied to the cactus model.

the model in order to test the robustness against the tracking assumption.

In the sequence of images in the figure 5, we compare the convergence behavior of a normal ICP algorithm against our structural variant when the tracking assumption is not met. For such cases, the pose estimation algorithm with the normal ICP variants does not converge to the actual model position. A direct comparison of the convergence behavior can be seen in the the first row of figure 6. Two different pose estimation algorithms were tested with our ICP variant, the 2D-3D (Rosenhahn and Sommer, 2005b) and projective ones (Araujo et al., 1998). In both cases, the structural ICP variants needs less iterations to converge.

The normal variants of the ICP algorithm consider as a correspondence constraint only the Euclidean distance plus a weighting error factor or a different search strategy. This has the effect that, in the first iterations many bad conditioned correspondences are found and therefore the convergence is slower or in some cases, the algorithm does not converge at all. The structural variant will also consider the constrains of equations (6), (8) and (7). This increases the probability to find better conditioned correspondences and therefore the convergence rate of the algorithm is increased.

A second experiment was made to test the robustness of our algorithm against the tracking assumption. For this case, the model was rotated around its z axis for zero to 50 degrees. As can be seen in the second row of figure 6, with the structural ICP algorithm the pose error is minimal for rotations up to 30 degrees for
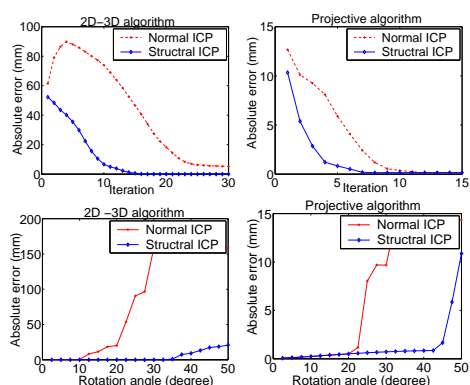
Figure 6: First row, convergence behavior comparisons of the normal and structural ICP variants applied to the 2D-3D pose estimation algorithm (left) and the projective algorithm (right). Second row, robustness against rotations for the 2D-3D (left) and the projective algorithms (right).

Figure 7: Pose error for the translation case for the cactus model (top left), for the puzzle model (top right) and for the mouse model (bottom).

the 2D-3D algorithm and 40 degrees for the projective one. This shows that our structural ICP variants allows larger model rotations than the normal ICP variants. The robustness of the structural ICP algorithm against the tracking assumption depends on the nature of the object and its contour. For contours which are rich in structural information larger rotations and translations are allowed.

The next experiment was made to test the magnitude and direction of the maximal possible translations allowed for the ICP structural algorithm. In this case, as can be seen in figure 4, the object model was translated to all directions in the plane where it is defined. For every position the pose was calculated with the structural ICP algorithm and the projective pose estimation (Araujo et al., 1998).

The results for the cactus, puzzle and mouse models are shown in figure 7. These figures show the convergence regions of the algorithm when translations are applied. For the cactus, the algorithm is more sensitive to translations in *y* direction, which corresponds to translations in the major axis direction (see figure 4), while relatively large translations are allowed in *x* direction (minor axis direction). The same effect can be seen for the puzzle model. The figures show that for certain positions the correspondence search is better conditioned. As the translation increases, the probability to find more bad conditioned correspondences also increases and therefore the pose error. The puzzle model and the cactus are complex objects, with enough structure to deal with relatively large translations. The bottom figure shows the result for the mouse model. In this case the mouse model does not have much structural information. Therefore, as can be seen in the figure 7, for large translations the error
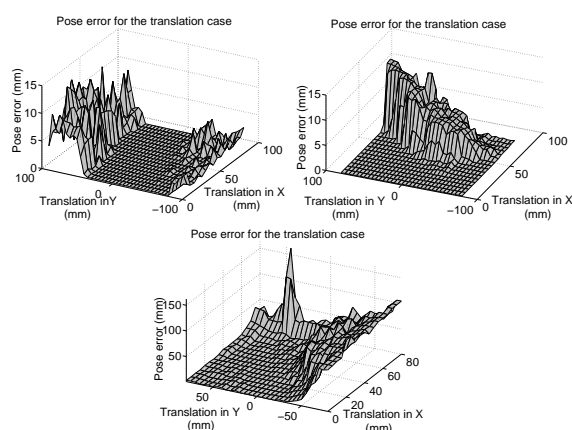
increases considerably.

Finally, we applied our algorithm to image sequences of a real scenario. The algorithm was tested on a Linux based system with a 3 Ghz. Intel Pentium 4 processor. Some examples of the test sequences are shown in the figure 8. The left column shows the initial position of our model and the right column the pose result using our ICP structural variant and the projective pose estimation algorithm. For every image the monogenic signal response was obtained and a contour search algorithm based on the local orientation and phase information was applied to detect the edge segments, then from these detected contour points the structural features were calculated. The average computing time per frame, for the hole image processing module, was 225 milliseconds. Due to the relatively large displacement of the object, more iteration steps are needed for the algorithm to converge and therefore the computational time increases. For these sequences the average computation time (image processing plus pose estimation) was 2.65 seconds.

6 CONCLUSIONS AND FUTURE WORK

A new variant of the ICP algorithm for pose estimation of 3D free-form contours based on local structural model and image features was presented. The experimental test proved that our structural ICP algorithm performs efficiently for rich structured objects, for large translations and rotations between scene and object model. The experiments show that our ICP algorithm combined with the projective pose estima-
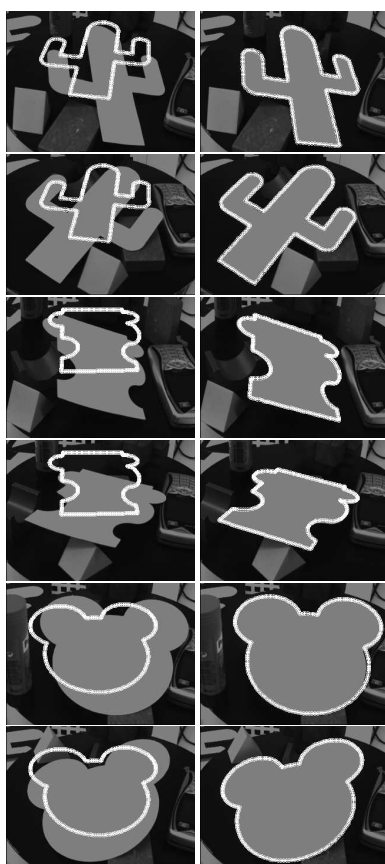
Figure 8: Initial position (left images) and estimated pose (right images) for the cactus, puzzle and mouse model.

tion approach can handle larger object displacements. That means, the feature constraints used to search correspondences and the pose estimation constraints involved in the minimization problem are better conditioned in the image plane. Although our approach does not reach requirements for real time applications (Rusinkiewicz and Levoy, 2001), the computation times reported for the test sequences are a good tradeoff if we consider that the tracking assumption has been significatively overcome. A natural extension for our approach is to consider the pose estimation of non-planar free-form contours and surfaces and to combine local and global structural features (from model and image) to develop an approach capable to deal with even larger translations and rotations.

REFERENCES

Araujo, H., Carceroni, R., and Brown, C. (1998). A fully projective formulation to improve the accuracy of Lowe's pose-estimation algorithm. *Comput. Vis. Image Underst.*, 70(2):227–238.

Benjemaa, R. and Schmitt, F. (1997). Fast global registration of 3d sampled surfaces using a multi-z-buffer technique. In *NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, page 113, Washington, DC, USA. IEEE Computer Society.

Besl, P. and McKay, N. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155.

Dorai, C., Weng, J., and Jain, A. (1997). Optimal registration of object views using range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1131–1138.

Felsberg, M. and Sommer, G. (2001). The monogenic signal. *IEEE Transactions on Signal Processing*, 49(12):3136–3144.

Felsberg, M. and Sommer, G. (2004). The monogenic scale-space: A unifying approach to phase-based image processing in scale-space. *J. Math. Imaging Vis.*, 21(1):5–26.

Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, 4(1):321–331.

Rosenhahn, B., Brox, T., Cremers, D., and Seidel, H. (2006). A comparison of shape matching methods for contour based pose estimation. In *11th International Workshop on Combinatorial Image Analysis (IWCIA)*. Berlin, Germany, LNCS Springer-Verlag.

Rosenhahn, B., Perwass, C., and Sommer, G. (2004). Free-form pose estimation by using twist representations. *Algorithmica*, 38:91–113.

Rosenhahn, B. and Sommer, G. (2005a). Pose estimation in conformal geometric algebra, part I: The stratification of mathematical spaces. *Journal of Mathematical Imaging and Vision*, 22:27–48.

Rosenhahn, B. and Sommer, G. (2005b). Pose estimation in conformal geometric algebra, part II: Real-time pose estimation using extended feature concepts. *Journal of Mathematical Imaging and Vision*, 22:49–70.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, Quebec City, Canada.

Shang, L., Jasiobedzki, P., and Greenspan, M. (2005). Discrete pose space estimation to improve ICP-based tracking. In *3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pages 523–530, Washington, DC, USA. IEEE Computer Society.

Sommer, G., editor (2001). *Geometric Computing with Clifford Algebras*. Springer-Verlag, Heidelberg.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, 13(2):119–152.