Three-Dimensional Shape Knowledge for Joint Image Segmentation and Pose Tracking

Bodo Rosenhahn[1] Thomas Brox[2]
Joachim Weickert[2]

[1] Centre for Imaging Technology and Robotics (CITR),
University of Auckland, New Zealand
bros028@cs.auckland.ac.nz

[2] Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science,
Saarland University, Building 27, 66041 Saarbrücken, Germany
{brox,weickert}@mia.uni-saarland.de

Abstract

In this article we present the integration of 3-D shape knowledge into a variational model for level set based image segmentation and tracking. Given a 3-D surface model of an object that is visible in the image of one or multiple cameras calibrated to the same world coordinate system, the object contour extracted by the segmentation method is applied to estimate the 3-D pose parameters of the object. Vice-versa, the surface model projected to the image plane helps in a top-down manner to improve the extraction of the contour. While common alternative segmentation approaches, which integrate 2-D shape knowledge, face the problem that an object can look very differently from various viewpoints, a 3-D free form model ensures that for each view the model can fit the data in the image very well. Moreover, one additionally solves the higher level problem of determining the object pose in 3-D space. Due to the variational formulation, the approach clearly states all model assumptions in a single energy functional that is locally minimized by our method. Its performance is demonstrated by experiments with a monocular and a stereo camera system.

1 Introduction

Image segmentation and pose estimation are two principal problems in computer vision. Segmentation determines the location and shape of objects in the image plane, thereby performing a significant abstraction step from the raw pixel data to object regions. It is well-known and easy to imagine that higher level vision problems get much simpler and more reliable, if the area in the image occupied by the object is known. However, image segmentation is a difficult task and often fails for general images. The main reason for these failures is a violation of the model assumptions imposed for image segmentation. Due to noise, texture, shading, occlusion, or simply because the appearance of two objects is locally nearly the same, the image gray value or color is rarely sufficient to clearly separate objects from their background. A possible remedy is the supplement of additional information, such as texture and motion information, which greatly extends the number of situations where image segmentation can succeed [33, 8]. Nevertheless, image segmentation is a too high level task for purely image-driven methods to succeed. For segmenting general images, prior object knowledge and a basic understanding of the scene are necessary to reliably determine the object boundaries. For this reason, image segmentation in the sense of object contour extraction can only work generally in the bundle with other high level vision tasks.

One such task is pose estimation, in particular 2D-3D pose estimation. Given a learned 3-D object model, the task of pose estimation is to estimate a rigid motion which fits the 3-D object model to some 2-D image data [26], i.e., one basically searches for the optimum six pose parameters – three for the object's translation and three for its rotation. To this end, the matching of features known from the 3-D object model to corresponding 2-D features in the image is necessary.
There are several possible features that could be matched between 2-D and 3-D, the object can be modelled in various ways, and there a different possibilities to determine the pose from the matched features. Therefore, it is not surprising that many works on 2D-3D pose estimation can be found in the literature [43, 25]. Pioneering work was done by Lowe [31, 32] and Grimson [26]. Their methods have been extended to fully projective formulations [1], the use of Plücker lines has been suggested [49], and pure rigid bodies have been extended to kinematic chains [4]. In [52] a neural network based approach has been proposed. The used features for matching range from lines [2], to viewpoint dependent point features including vertices, t-junctions, cusps, three-tangent junctions, edge inflections, etc. [29], or multi-part curve segments [55]. Also real-time tracking of articulated objects has been achieved by using contours [23].
The strategy followed in this paper is based on a matching between the 3-D object surface and the object contour in the image [44]. In detail, we try to find a rigid motion that minimizes the error between the projected object surface and the region encircled by the contour in the image. Since the common role of image segmentation is exactly to extract the contour of objects in the image, this shows the possible connection between 2D-3D pose estimation and image segmentation.

So image segmentation can serve the pose estimation task, yet what about the information flow in opposite direction? It has already been stated above that purely image driven segmentation methods are not able to extract the object contour in general sit-

uations. Hence, in some recent segmentation approaches, prior 2-D shape information has been integrated in order to impose additional constraints that force the contour to a desirable solution. An early example can be found in [30] where shape information influences the evolution of an active contour model. This basic concept has been extended and modified in [17, 47, 14, 20, 41, 19, 16] and provides a good framework for the sound integration of 2-D shape knowledge in segmentation processes.
However, the real world has three spatial dimensions. This fact is responsible for an inherent shortcoming of 2-D shape models: they cannot exactly describe the image of an object from arbitrary views. As a remedy to this problem, the different views of an object can be expressed by a statistical model [15]. The present paper instead embarks on the strategy to replace the 2-D shape model by a 3-D surface model, thus directly respecting the 3-D nature of the object.
For the integration of 3-D shape information in a 2-D segmentation process, the object model has to be projected onto the image plane, and for this its pose in the scene has to be known. At this point, one realizes again the connection between image segmentation and pose estimation, yet now the connection points into the other direction: a pose estimate is needed in order to integrate the surface model.
Notice that a pose estimation problem appears in the case of 2-D shape knowledge as well. Also there, it is necessary to estimate the translation, rotation, and scaling of the shape, before it can constrain the contour in the image. This is either achieved by explicit estimation of the pose parameters [47], or by an appropriate normalization of the shapes [16]. Extensions to perspective transformations of 2-D shapes have recently been proposed in [41]. However, all these efforts for estimating the 2-D pose only aim on the use of shape knowledge in order to yield improved segmentations. The 2-D pose estimates do not allow a location of the object in the real 3-D world but only in the 2-D projection of this world. In contrast, the 2D-3D pose estimation employed in our model not only helps to determine the object contour but also aims on the location of the object in the scene.

Bringing image segmentation and 2D-3D pose estimation together by formulating a joint energy functional is therefore a contribution that can be regarded from two perspectives. From the segmentation perspective, our approach extends methods that integrate 2-D prior knowledge to 3-D shape models with all its consequences. From the perspective of pose tracking, our method integrates the feature extraction step into the pose estimation process, i.e., there is a back-coupling of the pose result that helps to improve the extracted features, in our case the object silhouette.
Moreover, the method suggested in this paper is to our knowledge the first variational approach to pose estimation. Variational methods are very common in image analysis and belong to the best performing techniques, e.g., in image segmentation [8, 19], motion estimation [6], image denoising [48], or 3-D reconstruction [42, 53]. This is also because they provide a sound theoretical framework with all model assumptions clearly stated in a single energy functional and numerical schemes that provide at least a local optimum of this energy.
In the method described in this paper, the energy functional contains both the object contour and the pose parameters as unknowns. Since the optimum pose parameters depend on the contour and vice-versa, the minimization is done by alternating both image segmentation and pose estimation in an iterative manner, as illustrated in Figure 1.

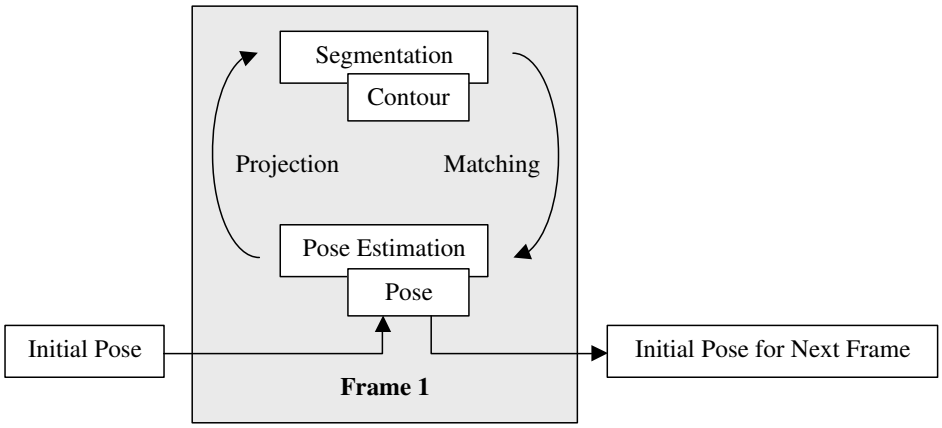

Figure 1: Basic idea: iterating segmentation and pose estimation. The projected pose result is used as a-priori knowledge for segmentation, the contour as matching feature for pose estimation.

This paper comprises and extends an earlier work presented on a conference [7]. In comparison to this introduction of the basic idea, the present paper contains a much more detailed description of the approach, integrates a confidence measure in the coupling of segmentation and pose estimation, and demonstrates the generality of the method by means of additional experiments that rule out many alternative techniques for solving the task.

Paper organization. The next section contains a detailed review of the level set based image segmentation model used in our approach. It further includes the introduction of a local region statistics model that aims on the handling of inhomogeneous objects and backgrounds. Section 3 then explains the concept of 2D-3D pose estimation including the contour based pose estimation technique needed for our approach. In Section 4 we then present our idea to combine image segmentation and 3-D pose estimation in a joint energy functional. Experiments in Section 5 demonstrate the performance of the proposed technique and illustrate the conceptual difference to other methods. The paper is concluded by a short summary in Section 6.

2 Image Segmentation

2.1 Level Set Formulation

Our method is based on variational image segmentation with level sets [22, 37, 11, 34, 38, 12], and in particular on the method described in [8, 5]. Level set formulations of the image segmentation problem have several advantages in comparison to other contour extraction methods. One of these advantages is the convenient embedding of a 1-D curve into a 2-D, image-like structure. This is very useful for the interaction between the constraints that are imposed on the contour itself and those constraints that act on the regions separated by the contour. A further advantage of level set segmentation is the capability of these methods to model topological changes of the regions. This can become very important, for instance, when the object is partially

occluded by another object and is hence split into two parts. Finally, level set methods easily allow the integration of further constraints like prior shape knowledge. Many recent methods that integrate shape knowledge in image segmentation are thus based on level set methods. In our approach we benefit from all these advantages. On the other hand, the most prominent drawback of level set methods, namely the difficulty to extend it to segmentations with more than two regions involved, is not important for the present method. This is because we only need a splitting of the image into the object, the pose of which we want to estimate, and its background. Thus a two-region segmentation is fully sufficient.

In level set based segmentation methods, a level set function $\Phi \in \Omega \mapsto \mathbb{R}$ splits the image domain Ω into two regions Ω_1 and Ω_2, with $\Phi(x) > 0$ if $x \in \Omega_1$ and $\Phi(x) < 0$ if $x \in \Omega_2$. The zero-level line thus marks the boundary between both regions, i.e., it represents the object contour that is sought to be extracted.

As an optimality criterion for the contour extraction, three constraints are imposed:

1. the data within each region should be as similar as possible
2. the data between regions should be as dissimilar as possible
3. the contour dividing the regions should be as short as possible

These model assumptions can be expressed by the following energy functional [56, 13]:

$$E(\Phi) = -\int_\Omega \big(H(\Phi)\log p_1 + (1-H(\Phi))\log p_2\big)\,dx + \nu \int_\Omega |\nabla H(\Phi)|\,dx \quad (2.1)$$

where $\nu > 0$ is a weighting parameter between the third and the two other constraints, and $H(s)$ is a regularized Heaviside function with $\lim_{s\to-\infty} H(s) = 0$, $\lim_{s\to\infty} H(s) = 1$, and $H(0) = 0.5$ (e.g. the error function). It indicates to which region a pixel belongs. Minimizing the first two terms maximizes the total a-posteriori probability given the probability densities p_1 and p_2 of Ω_1 and Ω_2, i.e., pixels are assigned to the most probable region according to the Bayes rule. The third term minimizes the length of the contour.

Energy minimization can be performed according to the gradient descent equation

$$\partial_t \Phi = H'(\Phi)\left(\log\frac{p_1}{p_2} + \nu\,\mathrm{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)\right) \quad (2.2)$$

where $H'(s)$ is the derivative of $H(s)$ with respect to its argument (so in our case a Gaussian). Applying this evolution equation to some initialization Φ^0, the contour converges to a (local) minimum for the numerical evolution parameter $t \to \infty$. This is illustrated in Figure 2.

2.2 Region Statistics

A very important factor for the quality of the contour extraction process is the way how the probability densities p_1 and p_2 are modelled. This model decides on what is considered as similar or dissimilar. There are several choices on which image cues to

use for the density model, for instance, gray value, color, texture [50, 39, 45], or motion [8, 40, 18]. Moreover, there are various possibilities how to model the probability densities given these image cues, e.g., a Gaussian density with fixed standard deviation [13], a full Gaussian density [46], a generalized Laplacian [27], or nonparametric Parzen estimates [28, 45, 8].

For the segmentation here, we use the texture feature space proposed in [9], which yields $M = 5$ feature channels I_j for gray scale images, and $M = 7$ channels if color is available. The color channels are considered in the CIELAB color space. The texture features described in [9] contain basically the same information as the frequently used responses of Gabor filters, yet the representation of this information is less redundant, so 4 feature channels substitute 12-64 Gabor responses.

The probability densities of the M feature channels are assumed to be independent, thus the total probability density comes down to

$$p_i = \prod_{j=1}^{M} p_{ij}(I_j) \qquad i = 1, 2. \tag{2.3}$$

Though assuming independence of the probability densities is only an approximation of the true densities, it keeps the density model tractable. This has to be seen particularly with regard to the fact that the densities have to be estimated by means of a limited amount of image data given.

Estimating both the probability densities p_{ij} and the region contour works according to the *expectation-maximization principle* [21, 35]. Having the level set function initialized with some partitioning, the probability densities can be approximated within the regions, for instance, by a Gaussian density estimate

$$p_{ij}(s) \propto \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\left(\frac{(s - \mu_{ij})^2}{2\sigma_{ij}^2}\right) \tag{2.4}$$

defined by the means μ_{ij} and standard deviations σ_{ij} in each region $i \in \{1, 2\}$ and channel $j \in \{1, ..., M\}$. This model contains only $2M$ parameters that have to be

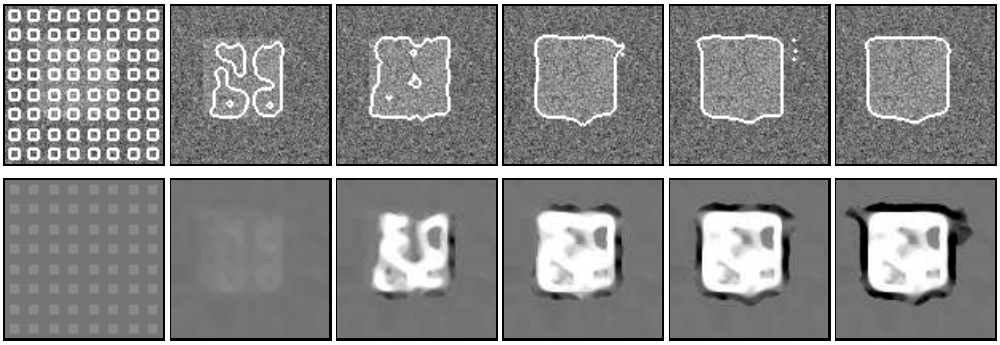

Figure 2: Illustration of the curve evolution according to (2.2). **From Left to Right:** Curve after $t = 0$, $t = 1$, $t = 2$, $t = 3$, $t = 4$, and $t = 20$. **Bottom Row:** Level set function Φ.

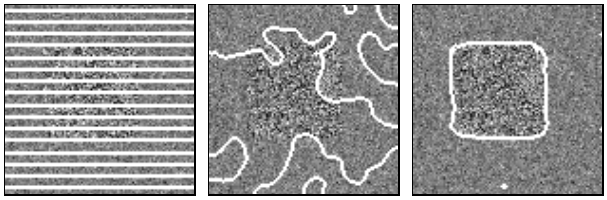

Figure 3: Synthetic image where regions have the same mean but different standard deviations. **From Left to Right:** Initialization. Result with the piecewise constant model. Result with the Gaussian model. Only the gray level information has been used for this illustration.

estimated within a region. In contrast to the Chan-Vese model, where only the means play a role [13], it has the advantage that the discrimination performance is independent of the contrast of a feature channel [5]. This means in particular that there is no need for explicitly weighting the different feature channels. Furthermore, regions can also be distinguished if they only differ in their standard deviations and not in their means; see Figure 3.

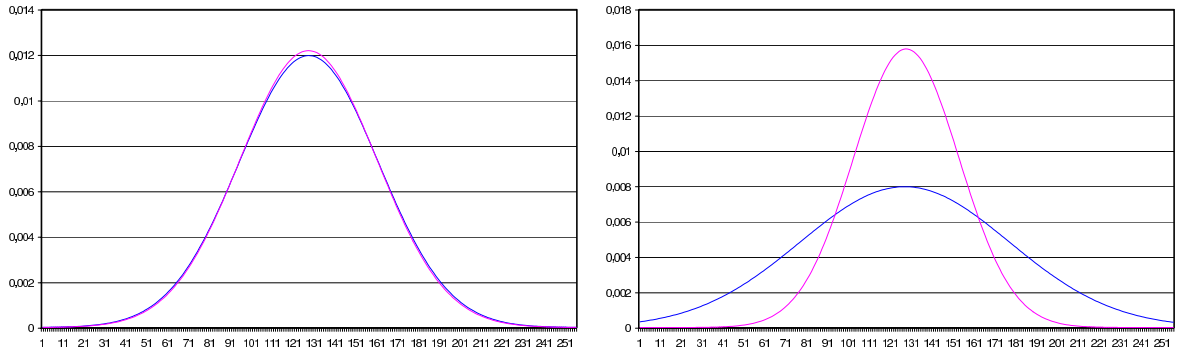

Figure 4: Illustration of how the Gaussian probability densities of the two regions are adapted due to the evolution of the contour in Figure 3. **Left:** Probability densities p_1 and p_2 estimated with the initial contour. **Right:** Probability densities p_1 and p_2 estimated with the final contour. It can be seen that the method maximizes the distance between the densities. The very small difference at the beginning is already sufficient to separate the regions.

With an approximation of the probability densities within the regions, one can compute an update on the contour according to (2.2), leading to a further update of the probability densities, and so on. Figure 4 illustrates the effect of this iterative process on the probability density estimates. It can be seen that the method maximizes the distance between the densities. Since the process converges to a local minimum, the initialization matters. In order to attenuate the dependency on the initialization, one can apply a continuation method in a coarse-to-fine manner [3].

Figure 5: Motivation for using local statistics. **Left:** In the global scope it is not clear whether the nose should be assigned to the zebra region or to the background (note the dark shadow). **Right:** When considering a local neighborhood, the assignment of the nose becomes unambiguous.

2.3 Local Region Statistics

The above-mentioned statistical model is a *global* model for the probability density of each region. Especially in complicated scenes with many objects in the background, shadows, and highlights, such a model may not be sufficient for separating the object region from the background. In such cases, differences between the regions are often only locally visible. A global statistical model looses this local information and can thus loose the capability to separate the regions. We propose here a remedy to this problem by suggesting *local* probability density estimates, where the densities may change with the position x in the image

$$p_{ij}(s,x) \propto \frac{1}{\sqrt{2\pi}\sigma_{ij}(x)} \exp\left(\frac{(s-\mu_{ij}(x))^2}{2\sigma_{ij}(x)^2}\right). \tag{2.5}$$

The parameters $\mu_{ij}(x)$ and $\sigma_{ij}(x)$ are computed in a local Gaussian neighborhood K_ρ around x by:

$$\mu_{ij}(x) = \frac{\int_{\Omega_i} K_\rho(\zeta - x) I_j(\zeta)\, d\zeta}{\int_{\Omega_i} K_\rho(\zeta - x)\, d\zeta} \qquad \sigma_{ij}(x) = \frac{\int_{\Omega_i} K_\rho(\zeta - x)(I_j(\zeta) - \mu_{ij}(x))^2\, d\zeta}{\int_{\Omega_i} K_\rho(\zeta - x)\, d\zeta} \tag{2.6}$$

where ρ denotes the standard deviation of the Gaussian window. In order to obtain reliable estimates for the parameters $\mu_{ij}(x)$ and $\sigma_{ij}(x)$, it is recommended to choose $\rho \geq 6$.

It should not be concealed that two major drawbacks come along with these local statistics. Firstly, they demand a considerably larger amount of computation time than global estimates. Secondly, they induce more local minima in the energy functional. The latter drawback is less severe for the approach discussed in the present paper, as the model uses object knowledge which constrains the subset of possible segmentations and provides a reasonable initialization for the contour. The first drawback, however, persists. Although one only has to compute the densities within the narrow band along the region boundary, the contour evolution using local statistics is about one order of magnitude slower than the same evolution with global statistics.

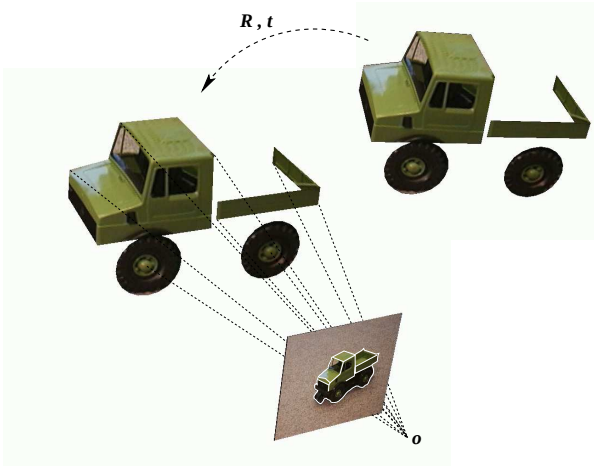

Figure 6: The pose scenario: the aim is to estimate the pose $\boldsymbol{R}$, $\boldsymbol{t}$.

3 2D-3D Pose Estimation

2D-3D pose estimation [26] means to estimate a rigid body motion which maps a 3D object model to an image of a calibrated camera, see Fig. 6. Depending on the camera model being used (orthographic, perspective), the object representation (e.g. point sets, line sets, contours, surface patches) and image data (e.g. corners, line segments, silhouettes) many different algorithms can be developed for different numerical estimation techniques (e.g. Kalman filters, gradient descent approaches, SVD decompositions), see [43, 25] for overviews. In this section, we summarize basic notations and previously developed point-based, contour-based, and surface-based pose estimation algorithms, see [43].

3.1 Foundations

We start with mathematic concepts that are needed for the pose problem, such as Plücker lines and twists to model rigid body motions. Then point-based and contour-based pose estimation algorithms are introduced.

3.1.1 Plücker lines

A 3-D line $\boldsymbol{L}$ can be represented in Plücker form [43]. A Plücker line $\boldsymbol{L} = (\boldsymbol{n}, \boldsymbol{m})$ is given as (unit) vector $\boldsymbol{n}$ and moment $\boldsymbol{m} = \boldsymbol{x} \times \boldsymbol{n}$ for a given point $\boldsymbol{x}$ on the line. An advantage of this representation is its uniqueness (apart from possible sign changes). This can be seen as follows: Let $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{L}$ with $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$. We choose $\lambda \in \mathbb{R}$ with $\boldsymbol{x}_2 = \lambda \boldsymbol{n} + \boldsymbol{x}_1$. Then we have (note: $\boldsymbol{n} \times \boldsymbol{n} = 0$)

$$\boldsymbol{x}_2 \times \boldsymbol{n} \;\; = (\lambda \boldsymbol{n} + \boldsymbol{x}_1) \times \boldsymbol{n} \;\; = \lambda(\boldsymbol{n} \times \boldsymbol{n}) + \boldsymbol{x}_1 \times \boldsymbol{n} = \boldsymbol{x}_1 \times \boldsymbol{n}. \tag{3.7}$$

Moreover, the incidence of a point $\boldsymbol{x}$ on a line $\boldsymbol{L} = (\boldsymbol{n}, \boldsymbol{m})$ can be expressed as

$$\boldsymbol{x} \in \boldsymbol{L} \;\; \Leftrightarrow \;\; \boldsymbol{x} \times \boldsymbol{n} - \boldsymbol{m} = 0. \tag{3.8}$$

This follows by a similar algebraic operation as in Equation 3.7.

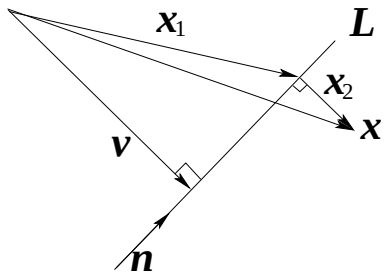

Figure 7: Comparison of a 3-D point with a 3-D line.

As a third advantage, Equation 3.8 provides us with a distance measure. Let $\boldsymbol{L} = (\boldsymbol{n}, \boldsymbol{m})$, with $\boldsymbol{m} = \boldsymbol{v} \times \boldsymbol{n}$ as shown in Figure 7, and $\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x_2}$, with $\boldsymbol{x} \notin \boldsymbol{L}$ and $\boldsymbol{x}_2 \perp \boldsymbol{n}$. Then we have (note: $\boldsymbol{x}_1 \times \boldsymbol{n} = \boldsymbol{m}$, $\boldsymbol{x}_2 \perp \boldsymbol{n}$ and $\|\boldsymbol{n}\| = 1$)

$$\|\boldsymbol{x} \times \boldsymbol{n} - \boldsymbol{m}\| \;=\; \|\boldsymbol{x}_1 \times \boldsymbol{n} + \boldsymbol{x}_2 \times \boldsymbol{n} - \boldsymbol{m}\| = \|\boldsymbol{x}_2 \times \boldsymbol{n}\| = \|\boldsymbol{x}_2\|. \tag{3.9}$$

This means that $\boldsymbol{x} \times \boldsymbol{n} - \boldsymbol{m}$ in Equation 3.8 results in a (rotated) perpendicular error vector to line $\boldsymbol{L}$. This distance measure and its minimization is used for pose estimation.

3.1.2 Rigid motions

Every 3-D rigid motion can be represented by a 4×4 matrix

$$\boldsymbol{M} \;=\; \begin{pmatrix} \boldsymbol{R}_{3\times3} & \boldsymbol{t}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 \end{pmatrix} \tag{3.10}$$

for a given rotation matrix $\boldsymbol{R}_{3\times3} \in SO(3)$, with $SO(n) := \{\boldsymbol{R} \in \mathbb{R}^{n\times n} : \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}, \quad det(\boldsymbol{R}) = +1\}$, and a translation vector $\boldsymbol{t}_{3\times1}$. By using homogeneous coordinates, a point $\boldsymbol{x}$ can be transformed by matrix multiplication $\boldsymbol{x}' = \boldsymbol{M}\boldsymbol{x}$. In fact, $\boldsymbol{M}$ is an element of the one-parametric Lie group $SE(3)$, known as the group of direct affine isometries. A main result of Lie theory is, that to each Lie group there exists a Lie algebra which can be found in its tangential space by derivation and evaluation at its origin; see [24, 36] for more details. The corresponding Lie algebra to $SE(3)$ is $se(3) = \{(\boldsymbol{v}, \omega) | \boldsymbol{v} \in \mathbb{R}^3, \omega \in so(3)\}$, with $so(3) = \{\boldsymbol{A} \in \mathbb{R}^{3\times3} | \boldsymbol{A} = -\boldsymbol{A}^T\}$. The elements in $se(3)$ are called *twists*, which can be denoted as

$$\hat{\xi} \;=\; \begin{pmatrix} \hat{\omega} & \boldsymbol{v} \\ 0_{3\times1} & 0 \end{pmatrix}, \text{ with } \hat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{3.11}$$

A twist is sometimes written as vector

$$\xi = (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3). \tag{3.12}$$

It contains six parameters and can be scaled to $\theta\xi$ for a unit vector ω. To reconstruct a group action $\boldsymbol{M} \in SE(3)$ from a given twist, the exponential function $\exp(\theta\hat{\xi}) = \boldsymbol{M} \in SE(3)$ can be used. The parameter $\theta \in \mathbb{R}$ corresponds to the motion velocity, i.e., the rotation velocity and pitch. For varying θ, the motion can be identified as screw motion

around an axis in space. This is proven by Chasles Theorem [36] from 1830. Indeed, evaluating the exponential of a matrix is not trivial, but it can be calculated efficiently by using the Rodriguez formula [36],

$$\exp(\hat{\xi}\theta) \;=\; \begin{pmatrix} \exp(\theta\hat{\omega}) & (I-\exp(\hat{\omega}\theta))(\omega\times\boldsymbol{v})+\omega\omega^T\boldsymbol{v}\theta \\ 0_{1\times 3} & 1 \end{pmatrix}, \text{ for } \omega\neq 0 \quad (3.13)$$

with $\exp(\theta\hat{\omega})$ computed by calculating

$$\exp(\theta\hat{\omega}) \;=\; I+\hat{\omega}\sin(\theta)+\hat{\omega}^2(1-\cos(\theta)), \quad (3.14)$$

i.e., only sine and cosine functions of real numbers need to be computed.

3.1.3 Point-based pose estimation

For point-based pose estimation we combine the results of both previous subsections and introduce a gradient descent method. The idea is to reconstruct an image point to a projection ray $\boldsymbol{L}=(\boldsymbol{n},\boldsymbol{m})$ and to claim incidence of the transformed 3-D point $\boldsymbol{x}$ with the 3-D ray:

$$(\exp(\theta\hat{\xi})\boldsymbol{x})_{3\times 1}\times\boldsymbol{n}-\boldsymbol{m}=0. \quad (3.15)$$

Indeed, $\boldsymbol{x}$ is a homogeneous 4-D vector, and after multiplication with the 4×4 matrix $\exp(\theta\hat{\xi})$ we neglect the homogeneous component (which is 1) to evaluate the cross product with $\boldsymbol{n}$. We now linearize the equation by using $\exp(\theta\hat{\xi})=\sum_{k=0}^{\infty}\frac{(\theta\hat{\xi})^k}{k!}\approx\boldsymbol{I}+\theta\hat{\xi}$, with $\boldsymbol{I}$ as identity matrix. This results in

$$((\boldsymbol{I}+\theta\hat{\xi})\boldsymbol{x})_{3\times 1}\times\boldsymbol{n}-\boldsymbol{m}=0 \quad (3.16)$$

and can be reordered into an equation of the form $\boldsymbol{A}\xi=\boldsymbol{b}$. Collecting a set of such equations (each is of rank two) leads to an overdetermined system of equations, which can be solved using, for example, the Householder algorithm. The Rodriguez formula can be applied to reconstruct the group action $\boldsymbol{M}$ from the estimated twist ξ. The group action is then applied to the 3-D points and the process is iterated until the gradient descent approach reaches a steady state.
Note that the projection rays only need to be reconstructed once, and can be reconstructed from orthographic, projective, or even catadioptric cameras. The algorithm is very fast (e.g., it needs 2 ms on a standard Linux PC for 100 point correspondences). In [43] extensions to point-plane, line-plane constraint equations and kinematic chains are presented using Clifford algebra [51].
In the described setting, the extension to multiple views is straightforward: we assume N images which are calibrated with respect to the same world coordinate system and are triggered. For each camera the system matrices $\boldsymbol{A}_1\ldots\boldsymbol{A}_N$ and solution vectors $\boldsymbol{b}_1\ldots\boldsymbol{b}_N$ are generated. The equations are now bundled in one system $\boldsymbol{A}=(\boldsymbol{A}_1,\ldots,\boldsymbol{A}_N)^T$ and $\boldsymbol{b}=(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_N)^T$. Since they are generated for the same unknowns ξ, they can be solved simultaneously, i.e., the spatial errors from all involved camera views are minimized.

3.1.4 Pose estimation of free-form contours

We assume a one-parametric closed curve in 3-D space,

$$C(\phi) \;=\; (f^1(\phi), f^2(\phi), f^3(\phi))^T, \tag{3.17}$$

which is represented by a finite set of contour points

$$C_(n) = \{(f^1(n), f^2(n), f^3(n))^T : n = 0 \ldots M-1\}. \tag{3.18}$$

The main idea is to interpret a one-parametric 3-D closed curve as three separate 1-D signals that 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 1-D discrete Fourier transform (1D-DFT). The inverse discrete Fourier transform (1D-IDFT) enables us to reconstruct low-pass approximations of each signal. Subject to the sampling theorem, this leads to the representation of the one-parametric 3-D curve $C(\phi)$ as

$$C(\phi) = \sum_{m=1}^{3} \sum_{k=-N}^{N} \boldsymbol{p}_k^m \exp\left(\frac{2\pi k\phi}{2N+1} i\right). \tag{3.19}$$

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. 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.
For pose estimation we combine this parametric representation within an iterated closest point (ICP) algorithm [54] in order to determine point correspondences between the image silhouette and the 3-D contour.

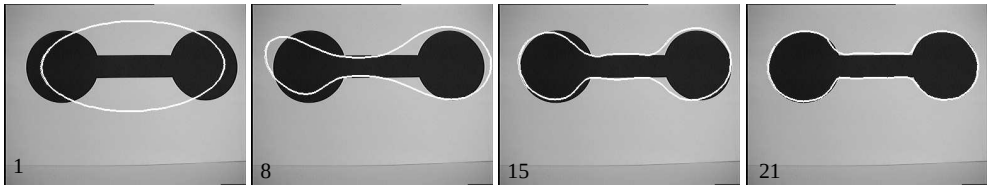

Figure 8: Pose results during iterated ICP-cycles and increased number of used Fourier descriptors.

The algorithm for pose estimation of free-form contours consists of iterating the following steps:

```
(a) Reconstruct the  projection rays from image points.
(b) Estimate the nearest point on the 3-D contour to each projection ray.
(c) Estimate the contour pose by using this point/line correspondence set.
(d) Goto (b).
```

For `(a)`, we determine from the calibration the optical center $\boldsymbol{c}$ and the image point in the world coordinate system. This is used to define the 3-D Plücker line, see section 3.1.1. For `(b)` we use Equation 3.7 and sample along the contour. Part `(c)` is described in Section 3.1.3. Figure 8 shows an example for iterations. The algorithm usually converges within 20 iterations and we need 40ms on a standard Linux PC (1GHz) to estimate the pose of a free-form contour.

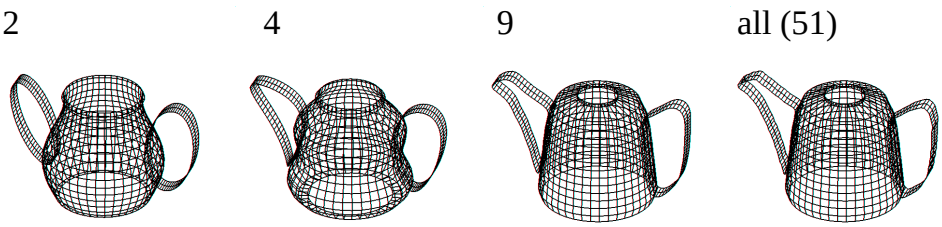

Figure 9: A sequence of different low-pass approximations of an object model consisting of three free-form surface patches.

3.1.5 Pose estimation of free-form surfaces

We assume a two-parametric surface [10] of the form

$$F(\phi_1, \phi_2) \quad = \quad (f^1(\phi_1, \phi_2), f^2(\phi_1, \phi_2), f^3(\phi_1, \phi_2))^T \tag{3.20}$$

defined by three functions $f^i(\phi_1, \phi_2) : \mathbb{R}^2 \rightarrow \mathbb{R}$ acting on the base vectors. The idea behind a two-parametric surface is to assume two independent parameters ϕ_1 and ϕ_2 which sample the 2-D surface in 3-D space. In fact, we are using a mesh model of the object [10]. For a finite 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 interpolate the surface by using a 2-D discrete Fourier transform (2D-DFT) and then apply an inverse 2-D discrete Fourier transform (2D-IDFT) for each base vector separately. Subject to a proper sampling, the surface can therefore be written as a series expansion of the form

$$\begin{aligned} F(\phi_1, \phi_2) \quad &= \quad \sum_{k_1=-N_1}^{N_1} \sum_{k_2=-N_2}^{N_2} \begin{pmatrix} F^1(k_1, k_2) \\ F^2(k_1, k_2) \\ F^3(k_1, k_2) \end{pmatrix} \exp\left(\frac{2\pi k_1 \phi_1}{2N_1 + 1} i\right) \exp\left(\frac{2\pi k_2 \phi_2}{2N_2 + 1} i\right) \text{ with} \\ F^j(k_1, k_2) \quad &= \quad \frac{1}{(2N_1 + 1)(2N_2 + 1)} \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} f^i(n_1, n_2) \\ &\quad \exp\left(-\frac{2\pi k_1 n_1}{2N_1 + 1} i\right) \exp\left(-\frac{2\pi k_2 n_2}{2N_2 + 1} i\right). \end{aligned} \tag{3.21}$$

Figure 9 shows approximation levels of a tea pot consisting of a handle, container and spout.

For pose estimation we assume a properly extracted silhouette of an object in a given image. There is a need to express tangentiality between the surface and the reconstructed projection rays, and there is also a need to express a distance measure within our description. For this requirement, we decided to use the 3-D rim of the surface model which is tangential with respect to the camera coordinate system. Here our tracking assumption comes into account: we project the 3-D surface with its initial pose onto a virtual image. Then the 2-D contour is calculated and from the image contour the 3-D rim of the surface model is reconstructed. To get the 3-D rim, there is a need to get from the image of a node point to its 3-D coordinates. This is done with the help of a look-up table F, see Figure 10. First we assume a mesh $C(i, j) \rightarrow (x, y, z)$ which gives the 3-D coordinates of the surface node for the two sample parameters (i, j). This mesh is projected with the projection matrix into a virtual image I. The model

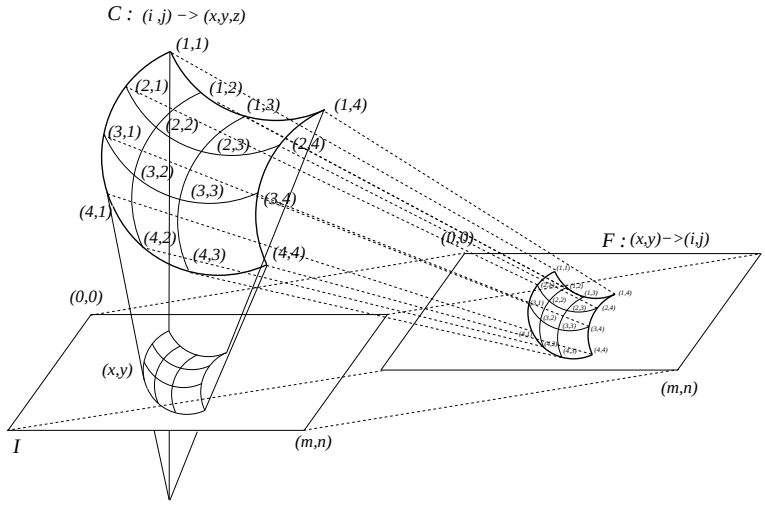

Figure 10: To determine the 3-D position of a 2-D image node, a field F is used as look-up table, which stores the relation between pixels and the 3-D mesh. $C(F(x, y))$ gives the 3-D coordinates of a node at pixel position (x, y) in the image.

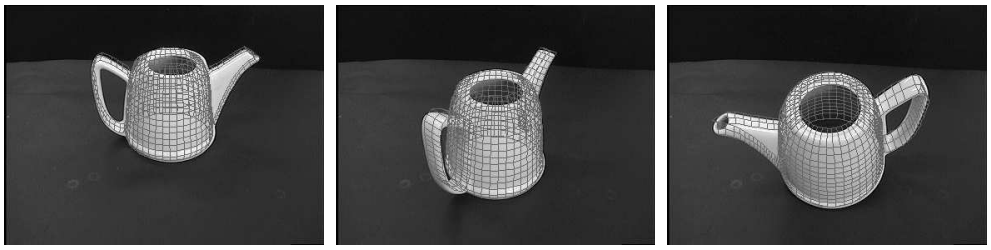

Figure 11: Pose results of the tea pot.

is projected in the virtual image with connecting line segments between points on the surface nodes and the nodes in another gray-scale value. This virtual image is used as a look-up table: we can detect the 3-D surface point for a given surface node on the image with the help of a 2-D field F and function C, since $C(F(x, y))$ yields the 3-D coordinates of the node's image point (x, y). To obtain the 3-D rim points we use a contour algorithm which follows the image of the mesh model by a recursive procedure. Then the nodes of the mesh model are collected, from which the corresponding 3-D rim is calculated with the help of F. The rim model is then applied on our contour-based pose estimation algorithm, see Section 3.1.4. Since the aspects of the surface model are changing during the ICP-cycles, a new rim will be estimated after each cycle. Pose results of the silhouette-based pose estimation algorithm are shown in Figure 11.
In order to deal with larger motions during pose tracking we use a sampling method that applies the surface based pose estimation algorithm for different neighboring starting positions. From all results we then choose the one with the minimum error between the extracted silhouette and the projected surface mesh.

4 Coupling Segmentation and Pose Estimation

The pose estimation method relies on the correctly extracted contour of the object in the image. Although the method can deal with some errors in the contour, it yields bad results as soon as the erroneous parts outnumber the correct parts. Therefore, the applicability of the pose estimation algorithm stands and falls with the capabilities of the contour extraction method, which must be able, e.g., to deal with texture.
On the other hand, the segmentation task can be simplified a lot, if the solutions are restricted to be close to a given prior shape. In many cases, this prevents the object region to capture parts of the image that do not belong to the object. Thus, using the known object shape from the 3-D model for segmentation and correcting its pose by the contour is beneficial both for the final contour and the estimated pose. In the following, we introduce a way how such a joint evolution of the contour and the pose can be realized in a variational setting.

4.1 Joint Energy Functional

In order to couple pose estimation and image segmentation in one single optimization problem, the energy functional for image segmentation in (2.1) is extended by an additional term that integrates the object model:

$$\begin{aligned} E(\Phi, \theta\xi) = \; & -\int_\Omega \big(H(\Phi)\log p_1 + (1-H(\Phi))\log p_2\big)\,dx + \nu \int_\Omega |\nabla H(\Phi)|\,dx \\ & + \underbrace{\lambda \int_\Omega (\Phi - \Phi_0(\theta\xi))^2\,dx}_{\text{Shape}}\,. \end{aligned} \tag{4.22}$$

The quadratic error measure in the shape term has been proposed in the context of 2-D shape priors, e.g. in [47]. The prior $\Phi_0 \in \Omega \to \mathbb{R}$ depends on the sought 3-D pose $\theta\xi$ and is assumed to be represented by the signed distance function. In our case this means, $\Phi_0(x)$ yields the distance of x to the silhouette of the projected object surface.

Assumed the pose parameters $\theta\phi$ are known, Φ_0 is constructed as follows: let X_S denote the set of points $\boldsymbol{X}$ on the object surface. Projection of the transformed points $\exp(\theta\xi)X_S$ into the image plane yields the set x_S of all 2-D points x on the image plane that correspond to a 3-D point on the surface model

$$x = P\exp(\theta\xi)\boldsymbol{X}, \qquad \forall \boldsymbol{X} \in X_S \tag{4.23}$$

where P denotes a projection with known camera parameters. The level set function Φ_0 can then be constructed from x_S by setting

$$\tilde{\Phi}_0(x) = \begin{cases} 1 & \text{if } x \in x_S \\ -1 & \text{otherwise} \end{cases} \tag{4.24}$$

and applying the distance transform, i.e.,

$$\Phi_0(x) = \begin{cases} \operatorname{dist}(x) & \text{if } \tilde{\Phi}_0(x) > 0 \\ -\operatorname{dist}(x) & \text{otherwise} \end{cases} \tag{4.25}$$

where $\operatorname{dist}(x)$ denotes the distance of x to the zero-level line of $\tilde{\Phi}_0$.

4.2 Energy Minimization

Minimizing (4.22) with respect to the contour Φ leads to the gradient descent equation

$$\partial_t \Phi \;=\; H'(\Phi)\left(\log\frac{p_1}{p_2} + \nu\,\mathrm{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)\right) + 2\lambda\,(\Phi_0(\theta\xi) - \Phi). \tag{4.26}$$

The shape term in this evolution equation pushes the contour towards the contour of the projected object model. This ensures that the contour cannot deviate too much from the modelled shape. The weighting parameter $\lambda \geq 0$ thereby determines just how far the contour can deviate from the prior. If the correct pose parameters were known, a large value of λ would ensure that the contour converges correctly to the shape of the object model.

However, the pose parameters are *not* known but are free variables and supposed to be optimized together with the contour. Thus the shape term in (4.22) not only draws the contour towards the projected object model, but also draws the object model towards the contour, or more precisely, makes the object model to change its pose such that the projected object Φ_0 resembles the contour Φ. This optimization of the pose parameters is achieved by the method described in Section 3. Due to the distance transform, the squared error measure in (4.22) is the error measure minimized by the least squares approach in Section 3: the minimum squared residual over all point-line correspondences is obtained by solving the overdetermined linear system $\mathbf{A}\xi = \mathbf{b}$ in Section 3.1.3.

In order to minimize the total energy, we suggest an iterative approach: keeping the contour Φ fixed, the optimum pose parameters $\theta\xi$ are determined as described in Section 3 and yield the silhouette Φ_0 of the object model. Retaining in the opposite way the pose parameters, (4.26) determines an update on the contour. Both iteration steps thereby minimize the distance between Φ and Φ_0. While the pose estimation method draws Φ_0 towards Φ, thereby respecting the constraint of a rigid motion, (4.26) in return draws the curve Φ towards Φ_0, thereby respecting the data in the image. The weighting parameter λ steers the influence of the object model versus the image data. A small λ gives the contour much freedom to evolve, enabling the pose to follow. If the object region can be clearly separated from the background, choosing λ small is therefore beneficial. On the other hand, if the contour is distracted by background clutter, the shape information keeps the contour from running too far away from the object. In our experiments we have chosen λ in the area of 0.05.

4.3 A Measure of Confidence for the Extracted Contour

The reliability of the extracted contour may vary a lot along the curve. Although the partitioning ensures a closed contour, which is in contrast to edge detection techniques, the separability of the object and the background region can be considerably reduced in some areas. This happens in particular in locations where the shape prior contradicts the local region statistics, e.g., due to occlusions. We therefore propose to measure the confidence of the extracted contour and to consider this information during pose estimation.

Figure 12: **From left to right:** (a) Initialization. (b) Segmentation result with object knowledge. (c) Pose result. (d) Segmentation result without object knowledge.

The sought confidence at a certain point x can be expressed by the probability that the point has been assigned to the correct region. This probability reads

$$\tilde{c}(x) = \frac{p_1(x) \int_{\Omega_1} K_\rho(x)\, d\xi}{p(x)} H(\Phi(x)) + \frac{p_2(x) \int_{\Omega_2} K_\rho(x)\, d\xi}{p(x)} (1 - H(\Phi(x))) \tag{4.27}$$

where K_ρ is the Gaussian kernel from Section 2.3. If a pixel assigned to region Ω_1 also fits well to region Ω_2, i.e., $p_1 \approx p_2$, the precise location of the contour will be ambiguous and the confidence will be around 0.5. Obversely, if a pixel assigned to Ω_1 does not fit to region Ω_2, i.e., $p_1 \gg p_2$, the contour location will be definite and the confidence will be close to 1. If a pixel is assigned to the wrong region according to the statistics – this can happen due to contradictions with the object prior or the length constraint – the confidence will be even smaller than 0.5.
Due to slightly blurred edges, pixels directly on the contour often have a quite low confidence, although the separability of the regions in the surrounding area is high. Therefore, it is reasonable to take also pixels from the neighborhood into account. This can be achieved by a simple convolution with a Gaussian kernel K_σ

$$c(x) = (K_\sigma * c)(x) \tag{4.28}$$

where we set $\sigma = 1.5$.

The confidence measure $c(x)$ can easily be integrated in the pose estimation procedure. Every equation in the linear system stemming from a correspondence of a 3-D point $\mathbf{X}$ with the 2-D point x is weighted by $c(x)$. This way, point matches in highly confident areas obtain more influence on the solution than matches stemming from ambiguous points on the contour.

5 Experiments

We investigated the performance of our joint contour extraction and pose estimation method in a couple of experiments. Fig. 12 first demonstrates the general advantage of integrating object knowledge into the segmentation process. Without object knowledge, parts of the tea box are missing as they better fit to the background. The object prior can constrain the contour to the vicinity of the projected object model derived from those parts of the contour that can be extracted reliably. This basic concept is also the key issue of approaches that use 2-D shape knowledge. With 3-D shape knowledge,

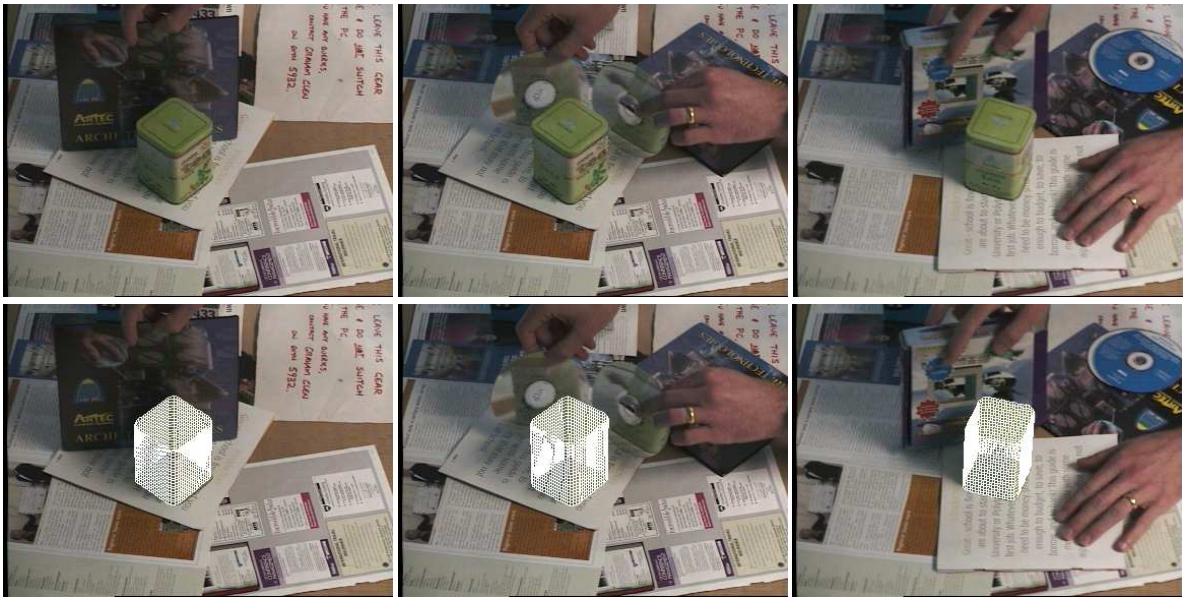

Figure 13: **Top row:** Input images for frames 51, 189 and 450 of an image sequence containing 560 frames. **Bottom row:** Pose results. The algorithm is able to deal with a cluttered and changing background.

however, it is no longer necessary to model several views. Moreover, the object model can perfectly fit the data, while in 2-D approaches there remain discrepancies if the current view does not coincide perfectly with one of the modelled views.
In Fig. 13 we show the robustness of the method in the case of a changing background. One can see that the estimated pose of the teabox is not distracted by any of the objects moved in the background, though the CDs even reflect the teabox surface. Later on in the sequence, also the teabox itself is moved, which shows that the method is not tuned for static objects.
In the experiment shown in Fig. 14, we tested the influence of artifacts like reflections, shadows, and noise. The motion of the object causes partially severe reflections on the metallic surface of the teabox. Moreover, the teabox throws a shadow as it is tilted. Additionally, Gaussian noise with standard deviation 30 has been added to the sequence. Nevertheless, the results remain stable. Also the slight occlusion due to the fingers does not harm the pose estimation. The presence of noise in this sequence clearly rules out methods that are based on background subtraction. Also simple thresholding methods for contour extraction would fail due to the cluttered background and the reflections.
Fig. 15 compares the results obtained with and without the suggested confidence measure, respectively. On the first glance, the improvement may not look extraordinary, yet the confidence measure prevents the result from being deteriorated by the shadow and the occluding fingers. With a homogeneous weighting, the correct contour points have not enough weight to ensure the correct pose estimate.
In the experiment depicted in Fig. 16, the monocular camera has been extended to a stereo system. In this case, another significant advantage of using 3-D shape knowledge becomes apparent. In contrast to 2-D approaches, our method can fuse the information from two images. If the information in one image is not reliable, e.g due to occlusions,

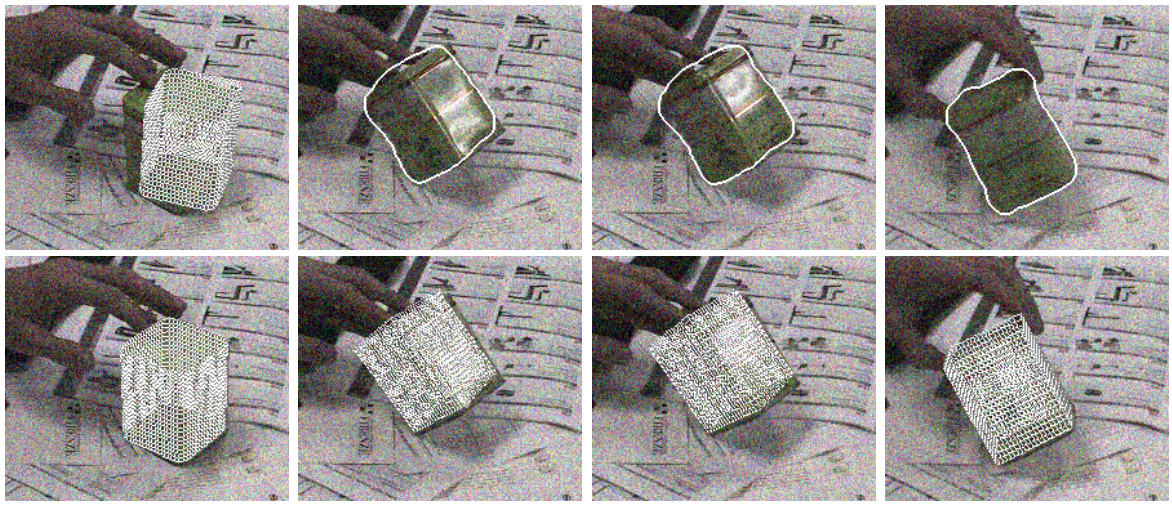

Figure 14: **Top row:** Initialization at the first frame. Contour at frames 49, 50, and 116 of the sequence. **Bottom row:** Pose results at frames 0, 49, 50, and 116. The teabox is moved, causing partially severe reflections on the box. Furthermore, Gaussian noise with standard deviation 30 has been added.

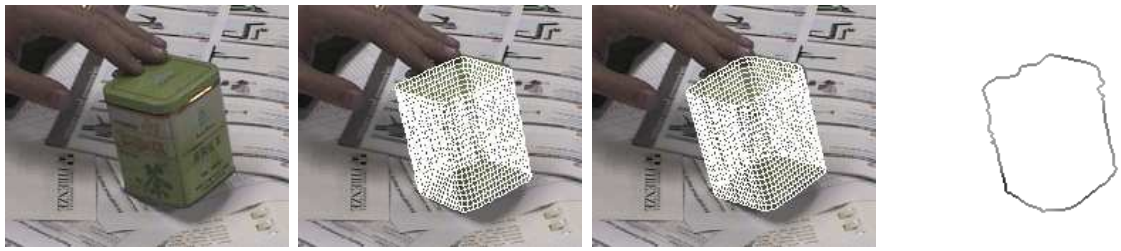

Figure 15: **From left to right:** (a) Frame 13. (b) Pose estimation result without the proposed confidence measure. (c) Result when exploiting the confidence. (d) Confidence along the contour. Dark values represent a high confidence.

the information from the other image can still determine the pose. Even if there are occlusions in both images, the combined information from both images can be still sufficient for a reliable pose estimation. The object model with the correct pose, on the other hand, constrains the contour and keeps it from breaking away.
In the sequel of this stereo sequence, the teabox is moved. Two further frames are depicted in Fig. 17. Again there appear reflections on the surface of the box, and there are further partial occlusions due to the hand.
In order to demonstrate that the approach is not restricted to a certain type of object, Fig. 18 shows an experiment with a teapot model. This object is non-convex and even contains a hole. Dealing with such a kind of object, it is particularly beneficial to represent the contour by means of a level set function. In the level set framework, the more complex topology does not change anything. Thus, the region encircled by the handle of the teapot can correctly be assigned to the background region. The teapot immediately rules out line-based methods for this task. Also methods based on feature matching may have difficulties due to the homogeneous surface of the object. Further note the bad initialization. A decoupled concatenation of the segmentation technique

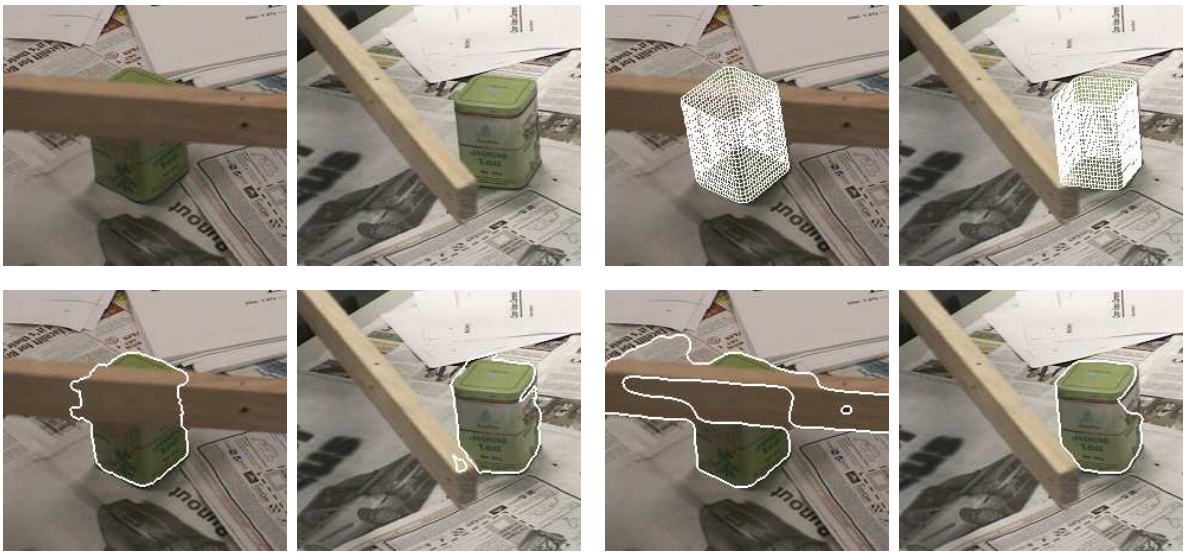

Figure 16: **Top left:** Frame 97 from a stereo sequence with 400 frames. **Top right:** Tracking result. In both views the object is partially occluded but the pose can be reconstructed thanks to the 3-D shape knowledge. **Bottom left:** The extracted contour is kept close to object. **Bottom right:** Here, the contour has been reinitialized at this frame with the correct contour, but the shape knowledge has been neglected. Consequently, the contour can break away.

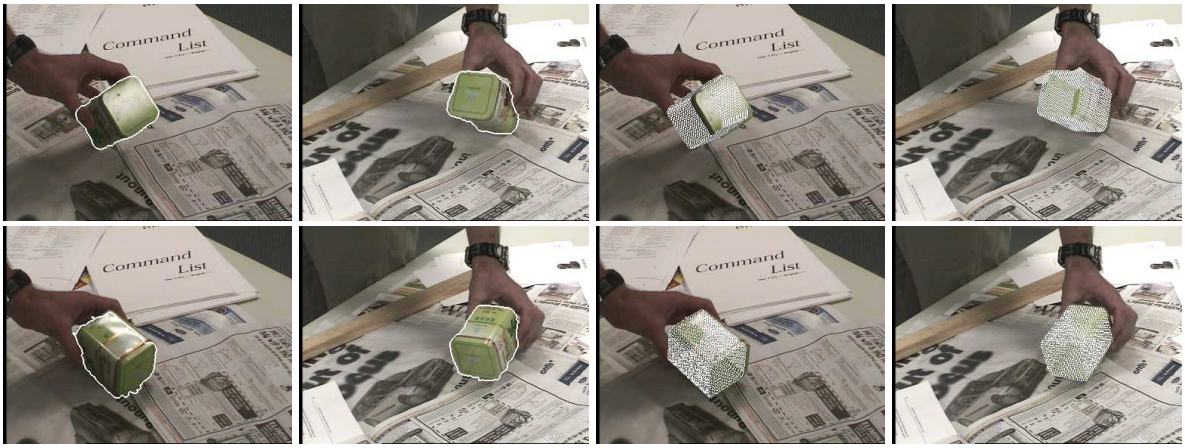

Figure 17: Contour and pose results at frame 190 and 212 for the stereo sequence from Fig. 16.

and the pose estimation method cannot succeed in finding the right contour and pose. Only the mutual improvement of both the contour and the pose allows for a good result in the steady state.

Finally, Fig. 19 depicts a sequence where object and camera are static to allow a quantitative error measurement. The two diagrams show the translational and angular errors along the three axes, respectively. Despite the change of the lighting conditions and partial occlusions, the error has a standard deviation of less than 5mm and 3.5 degree. The main rotational errors occur for rotations around the x-axis of the calibrated

Figure 18: **Top row:** Stereo image with a teapot. The initialization is quite far away from the object. **Center row:** Result in the steady state. The coupled contour extraction and pose estimation move the object prior into the right pose. **Bottom row:** Contour and pose with a simple concatenation of segmentation and pose estimation, i.e., only one iteration. The contour is restricted to the initial, bad pose and cannot fully capture the object. Consequently, the pose stays very close to the initialization.

system. This is due to the fact that such a rotation causes smaller changes of the silhouette than rotations around the other axes. Therefore, this degree of freedom is more sensitive to inaccuracies or errors in the extracted contour. The x-axis is located horizontally along the teapot, crossing the center of the teapot, and pointing from the handle to the spout.

6 Conclusion

In this work, variational and statistical methodologies have been combined with geometric techniques based on Clifford algebras. We introduced a method that integrates 3-D shape knowledge into a variational model for level set based image segmentation. While the utilization of 2-D shape knowledge has been investigated intensively in recent time, the presented approach takes the three-dimensional nature of the world into account. The method relies on a powerful image-driven segmentation model on one side, and

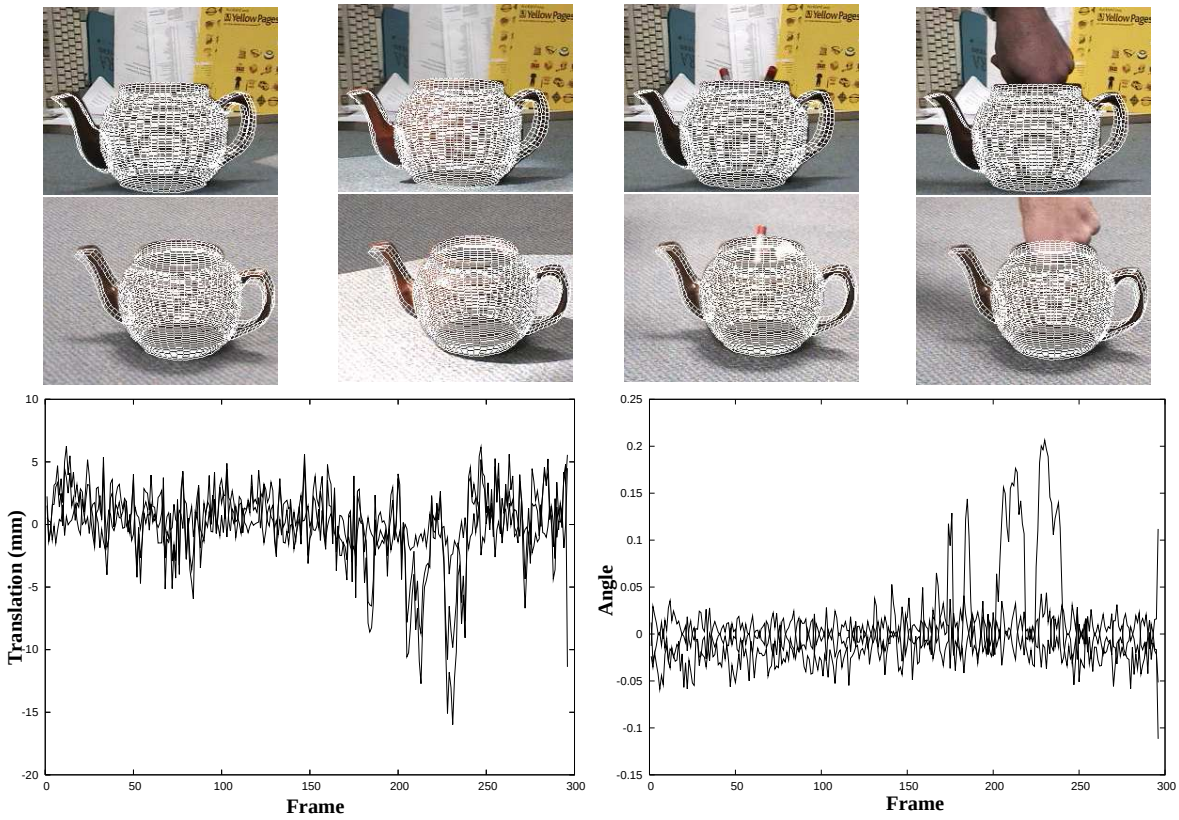

Figure 19: **Top rows:** Pose results at frame 30, 55, 130 and 200 for a static stereo sequence with illumination changes and partial occlusions (left and right view). **Bottom left:** Translational errors along the three coordinate axes in millimeter. **Bottom right:** Rotational errors in $\frac{180}{\pi}$ degree.

an elaborated technique for contour based 2D-3D pose estimation on the other side. The combination of both techniques improves the quality of contour extraction and, consequently, also the robustness of pose estimation that relies on the contour. This allows for the tracking of three-dimensional objects in cluttered scenes with inconvenient illumination effects and partial occlusions. The strategy to model the segmentation in the image plane, whereas the shape model is given in three-dimensional space, has the advantage that the image-driven part can operate on its natural domain as provided by the camera, while the 3-D object model offers the full bandwidth of perspective views. Moreover, in contrast to 2-D techniques, the object is given a location in space.

Acknowledgements

We gratefully acknowledge funding by the German Research Foundation (DFG) under the projects Ro2497/1 and We2602/1.

References

[1] H. Araújo, R. L. Carceroni, and C. M. Brown. A fully projective formulation to improve the accuracy of Lowe's pose-estimation algorithm. *Computer Vision and Image Understanding*, 70(2):227–238, May 1998.

[2] J. R. Beveridge. Local search algorithms for geometric object recognition: Optimal correspondence and pose. Technical Report Technical Report CS 93–5, University of Massachusetts, Amherst, 1993.

[3] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[4] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8–15, Santa Barbara, California, 1998.

[5] T. Brox. *From Pixels to Regions: Partial Differential Equations in Image Analysis*. PhD thesis, Faculty of Mathematics and Computer Science, Saarland University, Germany, Apr. 2005.

[6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision - Proc. 8th European Conference on Computer Vision*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36. Springer, Berlin, May 2004.

[7] T. Brox, B. Rosenhahn, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose estimation. In *Pattern Recognition*, Lecture Notes in Computer Science. Springer, Aug. 2005. To appear.

[8] T. Brox, M. Rousson, R. Deriche, and J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. In N. Petkov and M. A. Westenberg, editors, *Computer Analysis of Images and Patterns*, volume 2756 of *Lecture Notes in Computer Science*, pages 353–360. Springer, Berlin, Aug. 2003.

[9] T. Brox and J. Weickert. A TV flow based local scale estimate and its application to texture discrimination. *Journal of Visual Communication and Image Representation*, 2005. To appear.

[10] R. Campbell and P. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81:166–210, 2001.

[11] V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66:1–31, 1993.

[12] T. Chan and L. Vese. An active contour model without edges. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 141–151. Springer, 1999.

[13] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, Feb. 2001.

[14] D. Cremers. *Statistical Shape Knowledge in Variational Image Segmentation*. PhD thesis, Department of Mathematics and Computer Science, University of Mannheim, Germany, July 2002.

[15] D. Cremers, T. Kohlberger, and C. Schnörr. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36(9):1929–1943, Sept. 2003.

[16] D. Cremers, S. Osher, and S. Soatto. A multi-modal translation-invariant shape prior for level set segmentation. In C.-E. Rasmussen, H. Bülthoff, M. Giese, and B. Schölkopf, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 36–44. Springer, Berlin, Aug. 2004.

[17] D. Cremers, C. Schnörr, and J. Weickert. Diffusion-snakes: combining statistical shape knowledge and image information in a variational framework. In *Proc. First IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 137–144, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[18] D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005. To appear.

[19] D. Cremers, N. Sochen, and C. Schnörr. Multiphase dynamic labeling for variational recognition-driven image segmentation. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3024 of *Lecture Notes in Computer Science*, pages 74–86. Springer, Berlin, May 2004.

[20] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: introducing statistical shape knowledge into the mumford-shah functional. *International Journal of Computer Vision*, 50(3):295–313, Dec. 2002.

[21] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society series B*, 39:1–38, 1977.

[22] A. Dervieux and F. Thomasset. A finite element method for the simulation of Rayleigh–Taylor instability. In R. Rautman, editor, *Approximation Methods for Navier–Stokes Problems*, volume 771 of *Lecture Notes in Mathematics*, pages 145–158. Springer, Berlin, 1979.

[23] T. Drummond and R. Cipolla. Real-time tracking of multiple articulated structures in multiple views. In *Proc. 6th European Conference on Computer Vision, ECCV*, pages 20–36, Dublin, Ireland, 2000. Springer.

[24] J. Gallier. *Geometric Methods and Applications For Computer Science and Engineering.* Springer-Verlag, New York Inc., 2001.

[25] J. Goddard. *Pose and Motion Estimation From Vision Using Dual Quaternion-Based Extended Kalman Filtering.* PhD thesis, Knoxville, 1997.

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

[27] M. Heiler and C. Schnörr. Natural image statistics for natural image segmentation. *International Journal of Computer Vision*, 63(1):5–19, 2005.

[28] J. Kim, J. Fisher, A. Yezzi, M. Cetin, and A. Willsky. Nonparametric methods for image segmentation using information theory and curve evolution. In *IEEE International Conference on Image Processing*, volume 3, pages 797–800, Rochester, NY, June 2002.

[29] D. Kriegman, B. Vijayakumar, and J. Ponce. Constraints for recognizing and locating curved 3D objects from monocular image features. In G. Sandini, editor, *Proc. 2nd European Conference on Computer Vision (ECCV '92)*, volume 588 of *Lecture Notes in Computer Science*, pages 829–833. Springer, 1992.

[30] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 316–323, Hilton Head, SC, June 2000. IEEE Computer Society Press.

[31] D. Lowe. Solving for the parameters of object models from image descriptions. In *Proc. ARPA Image Understanding Workshop*, pages 121–127, 1980.

[32] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[33] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[34] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, Feb. 1995.

[35] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* Wiley series in probability and statistics. John Wiley & Sons, 1997.

[36] R. Murray, Z. Li, and S. Sastry. *Mathematical Introduction to Robotic Manipulation.* CRC Press, Boca Raton, FL, 1994.

[37] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[38] N. Paragios and R. Deriche. Unifying boundary and region-based information for geodesic active tracking. In *Proc. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 300–305, Forth Collins, Colorado, June 1999.

[39] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, Mar. 2002.

[40] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97(3):259–282, Mar. 2005.

[41] T. Riklin-Raviv, N. Kiryati, and N. Sochen. Unlevel-sets: geometry and prior-based segmentation. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3024 of *Lecture Notes in Computer Science*, pages 50–61. Springer, Berlin, May 2004.

[42] L. Robert and R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In B. Buxton and R. Cipolla, editors, *Computer Vision – ECCV '96*, volume 1064 of *Lecture Notes in Computer Science*, pages 439–451. Springer, Berlin, 1996.

[43] B. Rosenhahn. Pose estimation revisited. Technical Report TR-0308, Institute of Computer Science, University of Kiel, Germany, Oct. 2003.

[44] B. Rosenhahn and G. Sommer. Pose estimation of free-form objects. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3021 of *Lecture Notes in Computer Science*, pages 414–427, Prague, May 2004. Springer.

[45] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 699–704, Madison, WI, June 2003.

[46] M. Rousson and R. Deriche. A variational framework for active and adaptive segmentation of vector-valued images. In *Proc. IEEE Workshop on Motion and Video Computing*, pages 56–62, Orlando, Florida, Dec. 2002.

[47] M. Rousson and N. Paragios. Shape priors for level set representations. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision – ECCV 2002*, volume 2351 of *Lecture Notes in Computer Science*, pages 78–92. Springer, Berlin, 2002.

[48] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[49] F. Shevlin. Analysis of orientation problems using Plücker lines. In *International Conference on Pattern Recognition (ICPR)*, volume 1, pages 685–689, Brisbane, 1998.

[50] E. Sifakis, C. Garcia, and G. Tziritas. Bayesian level sets for image segmentation. *Journal of Visual Communication and Image Representation*, 13(1/2):44–64, 2002.

[51] G. Sommer, editor. *Geometric Computing with Clifford Algebra*. Springer Verlag, Berlin, 2001.

[52] S. Winkler, P. Wunsch, and G. Hirzinger. A feature map approach to real-time 3D object pose estimation from single 2D perspective views. In E. Paulus and F. Wahl, editors, *Mustererkennung 1997*, Informatik aktuell, pages 129–136. Springer, 1997.

[53] A. Yezzi and S. Soatto. Stereoscopic segmentation. In *Proc. 8th International Conference on Computer Vision*, volume 1, pages 59–66, Vancouver, Canada, July 2001.

[54] Z. Zang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1999.

[55] M. Zerroug and R. Nevatia. Pose estimation of multi-part curved objects. In *Proc. Image Understanding Workshop*, pages 831–835, 1996.

[56] S.-C. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, Sept. 1996.