SIAM J. IMAGING SCIENCES
Vol. 3, No. 1, pp. 110–132

A Geometric Approach to Joint 2D Region-Based Segmentation and 3D Pose Estimation Using a 3D Shape Prior*

Samuel Dambreville†, Romeil Sandhu‡, Anthony Yezzi‡, and Allen Tannenbaum‡

Abstract. In this work, we present an approach to jointly segment a rigid object in a two-dimensional (2D) image and estimate its three-dimensional (3D) pose, using the knowledge of a 3D model. We naturally couple the two processes together into a shape optimization problem and minimize a unique energy functional through a variational approach. Our methodology differs from the standard monocular 3D pose estimation algorithms since it does not rely on local image features. Instead, we use global image statistics to drive the pose estimation process. This confers a satisfying level of robustness to noise and initialization for our algorithm and bypasses the need to establish correspondences between image and object features. Moreover, our methodology possesses the typical qualities of region-based active contour techniques with shape priors, such as robustness to occlusions or missing information, without the need to evolve an infinite dimensional curve. Another novelty of the proposed contribution is to use a unique 3D model surface of the object, instead of learning a large collection of 2D shapes to accommodate the diverse aspects that a 3D object can take when imaged by a camera. Experimental results on both synthetic and real images are provided, which highlight the robust performance of the technique in challenging tracking and segmentation applications.

Key words. region-based segmentation and tracking, three-dimensional pose estimation, three-dimensional shape prior, variational methods, differential geometry

AMS subject classifications. 35-XX, 49-XX, 53-XX

DOI. 10.1137/080741653

1. Motivation and related work.

Two-dimensional (2D) image segmentation and 2D-3D pose estimation are key tasks for numerous computer vision applications and have received a great deal of attention in the past few years. These two fundamental techniques are usually studied separately in the literature. In this work, we combine both approaches in a single variational framework. To appreciate the contribution of this work, we recall some of the results and specifics of both fields.

2D-3D pose estimation aims at determining the pose of a 3D object relative to a calibrated camera from a single or a collection of 2D images. By knowing the mapping between the world

*Received by the editors November 24, 2008; accepted for publication (in revised form) November 12, 2009; published electronically March 3, 2010. A preliminary version of this paper appeared in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008 [24]. This work was supported in part by grants from NSF, AFOSR, ARO, and MURI, as well as by a grant from NIH (NAC P41 RR-13218) through Brigham and Women's Hospital. This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, grant U54 EB005149. Information on the National Centers for Biomedical Computing can be obtained from http://nihroadmap.nih.gov/bioinformatics.

http://www.siam.org/journals/siims/3-1/74165.html

†Corresponding author. School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 (samuel.dambreville@mba.gatech.edu).

‡School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 (rsandhu@gatech.edu, anthony.yezzi@ece.gatech.edu, tannenba@ece.gatech.edu).

coordinates and image coordinates from the camera calibration matrix, and after establishing correspondences between 2D features in the image and their 3D counterparts on the model, it is then possible to solve for the pose transformation parameters (from a set of equations that express these correspondences). The literature concerned with 3D pose estimation is very large, and a complete survey is beyond the scope of this paper. However, most methods can be distinguished by the type of *local* image features used to establish correspondences, such as points [1], lines or segments [2, 3], multipart curve segments [4], or complete contours [5, 6]. Segmentation consists of separating an object from the background in an image. The geometric active contour (GAC) framework, in which a curve is evolved continuously to capture the boundaries of an object, has proved to be quite successful at performing this task. Originally, the method focused on extracting local image features such as edges to perform segmentation; see [7, 8] and the references therein. However, edge-based techniques can suffer from the typical drawbacks that arise from using local image features: high sensitivity to noise or missing information, and a multitude of local minima that result in poor segmentations. Region-based approaches, which use global image statistics inside and outside the contour, were shown to drastically improve the robustness of segmentation results [9, 10, 11, 12]. These techniques are able to deal with various statistics of the object and background such as distinct mean intensities [10], Gaussian distributions [11, 12], or intensity histograms [13, 14, 15], as well as a wide variety of photometric descriptors such as grayscale values, color, or texture [16]. Further improvement of the GAC approach consists of learning the shape of objects and constraining the contour evolution to adopt familiar shapes to make up for poor segmentation results obtained in the presence of noise, clutter, or occlusion or when the statistics of the object and background are difficult to distinguish (see, e.g., [17, 18, 19, 20]).

1.1. Motivation/contribution. Our goal is to combine the strengths of both techniques (and to try to avoid some of their typical weaknesses) in order to both robustly segment 2D images and estimate the pose of an arbitrary 3D object whose shape is known.

In particular, we use a region-based approach to continuously drive the pose estimation process. This global approach avoids using local image features and, hence, addresses two shortcomings that typically arise from doing so in many 2D-3D pose estimation algorithms. First, finding the correspondence between local features in the image and on the model is a nontrivial task, due, for instance, to their viewpoint dependency—no local correspondences need to be found in our global approach. Second, local image features may not even exist or can be difficult to detect in a reliable and robust fashion in the presence of noise clutter or occlusion. Furthermore, simplifying assumptions usually need to be made on the class of shapes that a 2D-3D pose estimation technique can handle. Many approaches are limited to relatively simple shapes that can be described using geometric primitives such as corners, lines, circles, or cylinders. Recent work focused on free-form objects, which admit a manageable parametric description as in [5]. However, even this type of algebraic approach can become unmanageable for objects of arbitrary and complex shape. Our approach can deal with rigid objects of *arbitrary* shape, represented by a 3D level set [21] or a 3D cloud of points (see Figure 1).

Next, a major shortcoming of the GAC framework using shape priors is that 2D shapes are usually learned to segment 2D images. Hence, a large collection of 2D shapes needs to

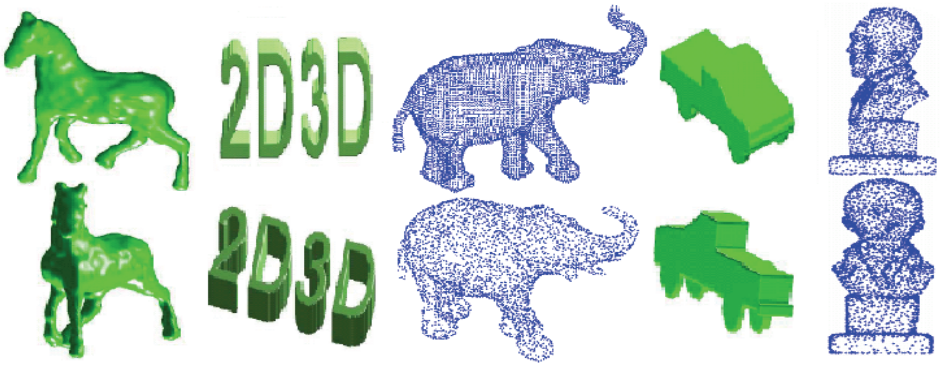


Figure 1. *Different views of the 3D models used in this paper (rendered surfaces or cloud of points).*

be learned in order to represent the wide variation in aspect that most natural 3D objects take when projected onto the 2D image plane. Our region-based approach benefits from the knowledge of the object shape that is compactly described by a *unique* 3D model. Acquisition of 3D models can be readily accomplished using range scans [22] or structure from motion approaches [23]. In addition, and in contrast to the GAC framework, the proposed method does not involve the evolution of an infinite dimensional contour to perform segmentation but only solves for the finite dimensional pose parameters (as is common for 2D-3D pose estimation approaches). This results in a much simplified framework that avoids dealing with problems such as infinite dimensional curve representation, evolution, and regularization.

1.2. Relation to previous work. In this paper, we expand the method presented in [24]. Our technique exploits many ideas from recent variational approaches that address the problem of structure from motion and stereo reconstruction from multiple cameras [25, 26, 23]. Originally, the authors in [26, 23] presented a method for reconstructing the 3D shape of an object from multiple 2D views obtained from calibrated cameras. The present contribution aims at performing a somewhat opposite task: given the 3D model of an object, perform the segmentation of 2D images and recover the 3D pose of the object relative to a *single* camera. To the best of our knowledge, this is the first time that the framework of [26, 23] has been adapted and employed in the specific context of segmenting 2D images from a single camera, using the knowledge of a 3D model. The framework in [23] has also recently been extended in [27] to address the problem of multiple camera calibration. In the present work, the camera is assumed to be calibrated. However, this assumption could easily be dropped by also solving for the optimal camera calibration parameters as presented in [27].

We note that, although the use of 3D shape knowledge to perform the 2D segmentation of regions presents obvious advantages, the literature dealing with this type of approach is strikingly thin. An early attempt to solve the problem of viewpoint dependency of the aspect of 3D objects can be found in [28]. In these papers, a region-based active contour approach is proposed that uses a unique shape prior. The prior shape is represented by a generalized cone based on *one* reference view of an object. The unlevel sections of the cone correspond to possible instances of the segmenting contour. Although the method performs well in the presence of variations in aspect of the object due to projective transformations, the method

cannot cope with images involving a view of the object that is radically different from the reference view. The closest piece of work to our proposed contribution is probably [29], which has been extended in [30]. In [29], the authors evolve an (infinite dimensional) active contour as well as 3D pose parameters to minimize a joint energy functional encoding both image information and 3D shape knowledge. Our method differs from the aforementioned approach in many crucial aspects. For example, we optimize a *single* energy functional, which allows us to circumvent the need to determine ICP-like[1] correspondences and to perform costly backprojections between the segmenting contour and the shape model at each iteration. Also, we perform optimization *only* in the finite dimensional space of the Euclidean pose parameters. In addition to being computationally efficient, this allows our technique to be less likely to be trapped in local minima, resulting in robust performances as demonstrated in the experimental part. In [30], the method of [29] is successfully simplified by eliminating the need to evolve an active contour and by performing energy minimization only in the space of 3D pose parameters. Thus, the method of [30] and our contribution present some similarities, notably in the use of the classical region-based energy functional introduced in [10] and [11]. However, the approaches to energy minimization and the resulting algorithms are radically different: In [30], an algebraic approach is used that involves establishing correspondences and backprojections between the 3D and 2D worlds, as well as linearizing the resulting system of equations. Consequently, important information about the geometry of the 3D model is lost through the algebraic approach. In contrast, our approach relies on surface differential geometry (see e.g., [31]) to link geometric properties of the model surface and its projection in the image domain. This allows us to derive the partial differential equations necessary to perform energy optimization. The resulting variational approach offers a complete and novel understanding of the problem of 3D pose estimation from 2D images. In addition, the knowledge of the 3D object is exploited to its full extent within our framework. In [32] the authors also successfully performed simultaneous 2D segmentation and 3D pose estimation using an entirely different approach. In their work, a cost function based on a Markov random field (MRF) was optimized using a dynamic graph cut approach (see [33] and the references therein). Also, and in contrast to our work, the 3D knowledge of the shape of an object was encoded via an articulated stick model instead of a 3D surface.

Our technique uses a 3D shape prior in a region-based framework and can thereby be expected to be robust to noise or occlusion. Hence, an obvious application of the proposed approach is the robust tracking of 3D rigid objects in 2D image sequences. Thus, our approach is also related to a wealth of methods concerned with the problem of model-based monocular tracking, one crucial difference being that most such approaches use local features in images (see [34] for a recent survey). In particular, in [35] a geometric approach to the 3D pose estimation problem is proposed: The authors use the knowledge of the occluding curve (i.e., the curve delimiting the visible part of the object from the camera) to search for edges in images and convincingly improve tracking performances. Similarly, the occluding curve plays a cornerstone role in our methodology.

This paper is organized as follows: In section 2, we detail our methodology by describing our choice of notation and energy functional, as well as by deriving the energy gradient to

[1]This refers to the Iterative Closest Point Algorithm.

solve the problem at hand. Then, we present experimental results for segmentation and tracking tasks that highlight the robustness of our technique to noise, clutter, occlusion, or poor initializations. Finally, we present our conclusions and future work.

2. Proposed approach. We suppose that we have at our disposal the 3D surface model of an object. Our goal is to find the 3D (Euclidean) transformation that needs to be applied to the model so that it coincides with the object of interest in the referential attached to a calibrated camera. To this end, we solve a typical shape optimization problem, in which we seek to segment the object in the 2D image plane with the 2D shape given by the projection of the 3D model for a given 3D transformation. The 3D transformation of the 3D model that results in an optimal segmentation of the object in the 2D image plane is expected to describe the actual position of the 3D object with respect to the camera. Therefore, the shape space (over which segmentation is performed) is the set of all 2D shapes determined by projection from the 3D model. This is a manifold, in which variational segmentation on the 3D transformation parameters can be performed. An overview of the method can be found in Figure 2. We now describe the proposed approach in detail, starting with our choice of notation.

2.1. Notation. Let $\mathbf{X} = [X, Y, Z]^T$ denote the coordinates of a point in $\mathbb{R}^3$, measured with respect to a referential attached to the imaging camera. We denote by I the image, by $\Omega \subset \mathbb{R}^2$ the image domain, and by $d\Omega$ its area element. We assume the camera is modeled as an ideal perspective projection:[2] $\pi : \mathbb{R}^3 \mapsto \Omega$; $\mathbf{X} \mapsto \mathbf{x}$, where $\mathbf{x} = [x, y]^T = [X/Z, Y/Z]^T$ denotes coordinates in Ω.

Let S be the smooth surface in $\mathbb{R}^3$ defining the shape of the object of interest. The (outward) unit normal to S at each point $\mathbf{X} \in S$ will be denoted by $\mathbf{N} = [N_1, N_2, N_3]^T$. To determine the pose of S with respect to the camera, we define the identical reference surface S_0, whose pose is known.[3] Denoting by X_0 the coordinates of points on S_0, one can locate S in the camera referential via the transformation $g \in SE(3)$, such that $S = g(S_0)$, or, written pointwise, $\mathbf{X} = g(\mathbf{X}_0) = \mathbf{R}\mathbf{X}_0 + \mathbf{T}$, with $\mathbf{R} \in SO(3)$ and $\mathbf{T} \in \mathbb{R}^3$. The parameters of the rigid motion g will be denoted by $\lambda = [\lambda_1, \ldots, \lambda_6]^T = [t_x, t_y, t_z, \omega_1, \omega_2, \omega_3]^T$ (where rotations are represented using exponential coordinates; see [38]).

Let $R = \pi(S) \subset \Omega$ be the region of the image on which the surface S projects (i.e., the region of Ω corresponding to imaging S). Let $R^c = \Omega \setminus R$ and $\hat{c} = \partial R$ denote the complement and the boundary of R, respectively (Figure 2). The curve $\hat{c} \subset \Omega$ is the projection of the curve $C \subset S$ that delineates the visible part of S from the camera: $\hat{c} = \pi(C)$. The 2D curve $\hat{c}$ and 3D curve C will be referred to, respectively, as the "silhouette" and the "occluding curve." The silhouette $\hat{c}$ will be parameterized by its arc-length $\hat{s}$. A point belonging to the silhouette will be denoted $\mathbf{y} = \mathbf{y}(\hat{s}) \in \hat{c}$. The (outward) normal to the curve $\hat{c}$ at $\mathbf{y}$ will be denoted $\hat{\mathbf{n}} = \hat{\mathbf{n}}(\mathbf{y})$. The occluding curve C will be parameterized by its arc-length s.

[2]More general models of cameras (see [36, 37]) can be straightforwardly handled. We make this assumption here to simplify the presentation.

[3]One can assume that the center of gravity of S_0 coincides with the camera center and that the rotation is known.

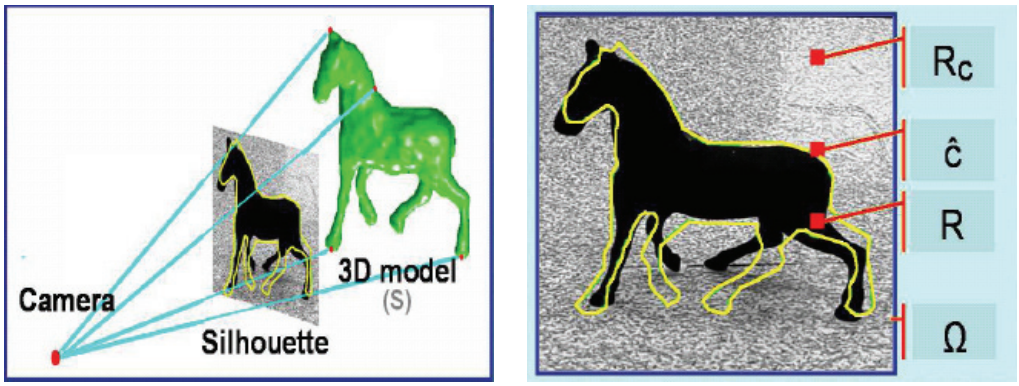


Figure 2. *Schema summarizing our segmentation/pose estimation approach from a 3D model. Left: First, the 3D model is transformed ($\mathbf{X} = g(\mathbf{X_0})$) and projected onto the 2D image plane ($\mathbf{x} = \pi(\mathbf{X})$). The resulting yellow curve is the "silhouette," i.e., the projection of the visible boundary of the 3D object onto the image plane. Right: Then, the gradient (for each parameter) is computed from the statistics inside (Region R) and outside (Region R^c) the silhouette $\hat{c}$.*

2.2. Energy functional. In [23], the authors employed an image formation approach to define a cost functional measuring the discrepancy between the photometric properties of the surface S (as well as the 3D background) and the pixel intensities of multiple images. The resulting energy involved backprojections to the surface S to guarantee the coherence between the measurements obtained from multiple cameras.

In the present work, we are interested in segmenting a *unique* image and we adopt a shape optimization approach, directly inspired from region-based active contours techniques [10, 11, 12, 13, 14]. Many segmentation approaches assume that the pixels corresponding to the object of interest or the background are distinct with respect to a certain grouping criterion. Within the GAC framework, region-based techniques perform segmentation by evolving a closed curve to increase the discrepancy between the statistics of the pixels located in the interior and exterior of the curve. Most region-based algorithms can be distinguished along three typical choices that are combined to separate the object from the background: The choice of the photometric variable (grayscale intensity, color, or texture vector), the choice of the statistical model for the photometric variables (probability density function), and the choice of the measure of similarity among distributions. These techniques minimize energies of the following form:

$$E = \int_{\mathrm{R}} r_{\mathrm{in}}\big(I(\mathbf{x}), \hat{c}\big) d\Omega + \int_{\mathrm{R^c}} r_{\mathrm{out}}\big(I(\mathbf{x}), \hat{c}\big) d\Omega, \tag{2.1}$$

where $r_{\mathrm{in}} : \mathcal{Z}, \Omega \mapsto \mathbb{R}$ and $r_{\mathrm{out}} : \mathcal{Z}, \Omega \mapsto \mathbb{R}$ are two monotonically decreasing functions measuring the matching quality of the image pixels with a statistical model over the regions R and R^c, respectively. The space $\mathcal{Z}$ corresponds to the photometric variable chosen to perform segmentation. Hence, depending on the choices for r_{in}, r_{out}, and $\mathcal{Z}$, a larger class of images than the ones fitting the specific hypotheses made in [23] can be dealt with.

The energy E measures the discrepancy between the statistical properties of the pixels located inside and outside the silhouette (curve $\hat{c}$) and does not involve any backprojections.

Although many measures of statistical similarity (e.g., Bhattacharyya distance as in [13] or mutual information as in [14]) could be chosen to define E, we use the log-likelihood function in this paper for simplicity.[4] Accordingly, one has

$$r_{\text{in}} = \log(P_{\text{in}}) \quad \text{and} \quad r_{\text{out}} = \log(P_{\text{out}}), \tag{2.2}$$

where P_{in} and P_{out} are the probability density functions (PDFs) of the pixels inside and outside the segmenting curve. We now detail possible choices of PDFs to model pixel statistics.

2.2.1. Gaussian assumption—identical variances. In [10], a method is proposed to segment images composed of regions of different mean intensities, using GACs. The resulting flow can be shown to be equivalent to comparing the log-likelihood of the Gaussian densities

$$P_{\text{in}}(I,\hat{c}) = \frac{1}{\sqrt{2\pi\Sigma_0}} e^{-\frac{(I-\mu_{\text{in}})^2}{2\Sigma_0}} \quad \text{and} \quad P_{\text{out}}(I,\hat{c}) = \frac{1}{\sqrt{2\pi\Sigma_0}} e^{-\frac{(I-\mu_{\text{out}})^2}{2\Sigma_0}}, \tag{2.3}$$

where the intensity averages of the pixels located inside and outside the curve $\hat{c}$ are denoted by μ_{in} and μ_{out}, respectively,[5] and $\Sigma_0 = \frac{1}{2}$. The averages μ_{in} and μ_{out} are computed at each step of the curve evolution as

$$\mu_{\text{in}}(\hat{c}) = \frac{\int_R I(\mathbf{x})\, d\Omega}{A_{\text{in}}} \quad \text{and} \quad \mu_{\text{out}}(\hat{c}) = \frac{\int_{R^c} I(\mathbf{x})\, d\Omega}{A_{\text{out}}} \tag{2.4}$$

with $A_{\text{in}}(\hat{c}) = \int_R d\Omega$ and $A_{\text{out}}(\hat{c}) = \int_{R^c} d\Omega$ the areas inside and outside the curve, respectively.

With the above notation and our particular choice of similarity measure and simplifying constant terms, the energy E can be defined as

$$r_{\text{in}} = -(I(\mathbf{x}) - \mu_{\text{in}})^2 \quad \text{and} \quad r_{\text{out}} = -(I(\mathbf{x}) - \mu_{\text{out}})^2. \tag{2.5}$$

2.2.2. Gaussian assumption—different variances. In [11, 39], a method is proposed to segment images composed of regions with distinct Gaussian densities, using the estimates

$$P_{\text{in}}(I,\hat{c}) = \frac{1}{\sqrt{2\pi\Sigma_{\text{in}}}} e^{-\frac{(I-\mu_{\text{in}})^2}{2\Sigma_{\text{in}}}} \quad \text{and} \quad P_{\text{out}}(I,\hat{c}) = \frac{1}{\sqrt{2\pi\Sigma_{\text{out}}}} e^{-\frac{(I-\mu_{\text{out}})^2}{2\Sigma_{\text{out}}}}, \tag{2.6}$$

where the variances of the pixels located inside and outside the curve $\hat{c}$ are denoted by Σ_{in} and Σ_{out}, respectively.[6] The variances Σ_{in} and Σ_{out} are supposed to be distinct.[7] The intensity averages μ_{in} and μ_{out} are computed as above, and the variances Σ_{in} and Σ_{out} are computed at each step of the curve evolution as

$$\Sigma_{\text{in}}(\hat{c}) = \frac{\int_R \left(I(\mathbf{x}) - \mu_{\text{in}}\right)^2 d\Omega}{A_{\text{in}}} \quad \text{and} \quad \Sigma_{\text{out}}(\hat{c}) = \frac{\int_{R^c} \left(I - \mu_{\text{out}}\right)^2 d\Omega}{A_{\text{out}}}. \tag{2.7}$$

[4] Moreover, intrinsic behaviors due to a particular choice of similarity measure that can be observed in the GAC framework, where an infinite dimensional curve is evolved, are likely to be less prominent in our particular framework where the shape of the segmenting curve can only be possible silhouettes of the 3D object.

[5] For grayscale images, $\mu_{\text{O/B}}$ are scalars. For color images, $\mu_{\text{O/B}} \in \mathbb{R}^3$.

[6] For grayscale images, $\Sigma_{\text{O/B}}$ is a scalar. For color images, $\Sigma_{\text{O/B}} \in \mathbb{R}^{3\times 3}$. Texture can also be used; see [16].

[7] The case where $\Sigma_{\text{in}} = \Sigma_{\text{out}}$ is treated as above.

In this case, with our particular choice of similarity measure and simplifying constant terms, the energy E can be defined as

$$r_{\text{in}} = -\log(\Sigma_{\text{in}}) - \frac{(I(\mathbf{x}) - \mu_{\text{in}})^2}{\Sigma_{\text{in}}} \quad \text{and} \quad r_{\text{out}} = -\log(\Sigma_{\text{out}}) - \frac{(I(\mathbf{x}) - \mu_{\text{out}})^2}{\Sigma_{\text{out}}}. \tag{2.8}$$

2.2.3. Using generalized distributions. The Gaussian models alluded to above can be too simplistic to accurately separate the object from the background. One solution is to use less constrained models of the distributions of the object and background, e.g., Parzen estimators and generalized histograms. This has been investigated in [13, 14] within the GAC framework, as well as in [15] within a model-based segmentation approach that also aimed at estimating the pose parameters of medical structures. In a similar manner, the PDFs P_{in} and P_{out} are computed from the silhouette as

$$P_{\text{in}}(z, \hat{c}) = \frac{\int_R \mathbf{K}(I(\mathbf{x}) - z)d\Omega}{A_{\text{in}}} \quad \text{and} \quad P_{\text{out}}(z, \hat{c}) = \frac{\int_{R^c} \mathbf{K}(I(\mathbf{x}) - z)d\Omega}{A_{\text{out}}} \tag{2.9}$$

with $\mathbf{K}(\chi)$ typically being a smooth version of the Dirac function, e.g., $\mathbf{K}(\chi) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\chi^2}{2\sigma^2}}$ for a sufficiently small value of σ.

2.3. Gradient flow. Following the region-based segmentation paradigm, the energy E is expected to be minimal when R and R^c correspond to the object and background in I, respectively. Most region-based approaches evolve an infinite dimensional curve, which amounts to exploring unconstrained shapes of the segmenting contour. Since we assume that the 3D shape of the rigid object is known, we want to minimize E by exploring only the possible regions R and R^c that result from projecting the surface S onto the image plane. For a calibrated camera, these regions are functions of the transformation g only. Solving for the transformation that minimizes E can be undertaken via gradient descent over the parameters λ, as described below.

The partial differentials of E with respect to the pose parameters λ_i's can be computed using the chain rule:

$$\begin{aligned} \frac{dE}{d\lambda_i} &= \int_{\hat{c}} \big(r_{\text{in}}\big(I(\mathbf{x})\big) - r_{\text{out}}\big(I(\mathbf{x})\big)\big) \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} \\ &\quad + \int_R \left\langle \frac{\partial r_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega + \int_{R^c} \left\langle \frac{\partial r_{\text{out}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega. \end{aligned} \tag{2.10}$$

The gradient in (2.10) involves the computation of the shape derivative $\frac{\partial \hat{c}}{\partial \lambda_i}$, which describes the directions of deformation of the 2D curve (under projection) with respect to the 3D pose parameter. The gradient is composed of three terms. The first is the dot product of a typical 2D region-based gradient (i.e., $(r_{\text{in}} - r_{\text{out}}).\hat{\mathbf{n}}$; see e.g., Chan and Vese's model [10]) with the shape derivative: for each point on the 2D curve, the deformation direction is compared to the normal, $\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \rangle$, and weights the statistical comparison term, $r_{\text{in}} - r_{\text{out}}$. The average over each point of the curve determines the optimal direction of variation of the pose parameter λ_i (i.e., the sign of the derivative $\frac{dE}{d\lambda_i}$). The two last terms simply measure the variation of the statistical measures r_{in} and r_{out} with the variation in pose.

In the remainder of this section, we first detail each of the three terms in (2.10) for the different statistical models presented above. Then we present further computations to express the gradient as a function of the known terms. Finally, we conclude the section by presenting remarks concerning the gradient and its implementation.

2.3.1. Gaussian assumption—identical variances (again). When the regions inside and outside the silhouette are modeled by Gaussian PDFs as in subsection 2.2.1, the second term in (2.10) may be calculated using the chain rule as

$$\begin{aligned}\int_R \left\langle \frac{\partial r_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega &= \int_R \left\langle 2\big(I(\mathbf{x}) - \mu_{\text{in}}\big) \frac{\partial \mu_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega \\ &= 2 \left\langle \frac{\partial \mu_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle \int_R (I(\mathbf{x}) - \mu_{\text{in}}) d\Omega \\ &= 2 \left\langle \frac{\partial \mu_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle \big[\mu_{\text{in}}.A_{\text{in}} - \mu_{\text{in}}.A_{\text{in}}\big] = 0.\end{aligned} \tag{2.11}$$

Similarly, the third term in (2.10) can also be shown to collapse. Hence, the partial derivative of (2.10) is simply

$$\frac{dE}{d\lambda_i} = \int_{\hat{c}} \big((I(\mathbf{y}) - \mu_{\text{out}})^2 - (I(\mathbf{y}) - \mu_{\text{in}})^2\big) \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s}. \tag{2.12}$$

2.3.2. Gaussian assumption—different variances (again). When the regions inside and outside the silhouette are modeled by Gaussian PDFs as in subsection 2.2.2, the second term in (2.10) may be computed using the chain rule as

$$\begin{aligned}&\int_R \left\langle \frac{\partial r_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega = \underbrace{\int_R \left\langle 2\left(\frac{I(\mathbf{x}) - \mu_{\text{in}}}{\Sigma_{\text{in}}}\right) \frac{\partial \mu_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega}_{=0 \ \text{(see above)}} \\ &- \int_R \left\langle \left(\frac{\Sigma_{\text{in}} - (I(x,y) - \mu_{\text{in}})^2}{\Sigma_{\text{in}}^2}\right) \frac{\partial \Sigma_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega \\ &= -\frac{1}{\Sigma_{\text{in}}^2} \left\langle \frac{\partial \Sigma_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle \int_R \big(\Sigma_{\text{in}} - (I(x,y) - \mu_{\text{in}})^2\big) d\Omega \\ &= -\frac{1}{\Sigma_{\text{in}}^2} \left\langle \frac{\partial \Sigma_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle (A_{\text{in}}\Sigma_{\text{in}} - A_{\text{in}}\Sigma_{\text{in}}) = 0.\end{aligned} \tag{2.13}$$

Similarly, the third term in (2.10) can also be shown to collapse. Hence, the partial derivative of (2.10) is simply

$$\frac{dE}{d\lambda_i} = \int_{\hat{c}} \left(\log\left(\frac{\Sigma_{\text{out}}}{\Sigma_{\text{in}}}\right) + \frac{(I(\mathbf{y}) - \mu_{\text{out}})^2}{\Sigma_{\text{out}}} - \frac{(I(\mathbf{y}) - \mu_{\text{in}})^2}{\Sigma_{\text{in}}}\right) \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s}. \tag{2.14}$$

2.3.3. Using generalized distributions (again). For generalized histograms as computed in (2.9) and using the chain rule, one can compute the second term of (2.10) as

$$\int_R \left\langle \frac{\partial r_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega = \int_R \left\langle \frac{1}{P_{\text{in}}(I(\mathbf{x}))} \frac{\partial P_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega. \tag{2.15}$$

Using the calculus of variations, one may derive that at a particular point $\mathbf{y} \in \hat{c}$

$$\frac{\partial P_{\text{in}}}{\partial \hat{c}}(z, \hat{c}) = \frac{\mathbf{K}(I(\mathbf{y}) - z) - P_{\text{in}}(z, \hat{c})}{A_{\text{in}}}\, \hat{\mathbf{n}}(\mathbf{y}).$$

Plugging this into (2.15), one gets

$$\int_R \left\langle \frac{\partial r_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega = \int_R \left(\int_{\hat{c}} \frac{\mathbf{K}(I(\mathbf{y}) - I(\mathbf{x})) - P_{\text{in}}(I(\mathbf{x}))}{P_{\text{in}}(I(\mathbf{x})).A_{\text{in}}} \left\langle \hat{\mathbf{n}}(\mathbf{y}), \frac{\partial \hat{c}}{\partial \lambda_i}(\mathbf{y}) \right\rangle d\hat{s} \right) d\Omega, \tag{2.16}$$

where we expressed the fact that the scalar product $\langle ., . \rangle$ in the left-hand side is a line integral on $\hat{c}$ (since $\frac{\partial r_{\text{in}}}{\partial \hat{c}}$ and $\frac{\partial \hat{c}}{\partial \lambda_i}$ are vector fields on $\hat{c}$). Swapping integrals (all integrations being done on compact sets), one can write

$$\int_R \left\langle \frac{\partial r_{\text{in}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega = \int_{\hat{c}} \mathcal{R}_{\text{in}}\left(I(\mathbf{y})\right) \left\langle \hat{\mathbf{n}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\hat{s} \tag{2.17}$$

with

$$\mathcal{R}_{\text{in}}(z) = \int_R \frac{\mathbf{K}(z - I(\mathbf{x})) - P_{\text{in}}(I(\mathbf{x}))}{A_{\text{in}}.P_{\text{in}}(I(\mathbf{x}))} d\Omega = \frac{1}{A_{\text{in}}} \int_R \frac{\mathbf{K}(z - I(\mathbf{x}))}{P_{\text{in}}(I(\mathbf{x}))} d\Omega - 1. \tag{2.18}$$

The third term of (2.10) can be computed in a similar fashion, yielding

$$\int_{R^c} \left\langle \frac{\partial r_{\text{out}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega = \int_{\hat{c}} \mathcal{R}_{\text{out}}\left(I(\mathbf{y})\right) \left\langle \hat{\mathbf{n}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\hat{s} \tag{2.19}$$

with

$$\mathcal{R}_{\text{out}}(z) = 1 - \frac{1}{A_{\text{out}}} \int_{R^c} \frac{\mathbf{K}(z - I(\mathbf{x}))}{P_{\text{out}}(I(\mathbf{x}))} d\Omega. \tag{2.20}$$

Hence, the partial derivative of (2.10) is simply

$$\frac{dE}{d\lambda_i} = \int_{\hat{c}} \left\{ r_{\text{in}} - r_{\text{out}} + \mathcal{R}_{\text{in}} + \mathcal{R}_{\text{out}} \right\} \left(I(\mathbf{y})\right) \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s}. \tag{2.21}$$

Note that when the Dirac function is used as the kernel $\mathbf{K}$ to compute P_{in} and P_{out} in (2.9), one can show that the terms $\mathcal{R}_{\text{in}}$ and $\mathcal{R}_{\text{out}}$ collapse (this is done using the sifting property of the Dirac function in (2.18) and (2.20)).

2.3.4. Making the gradient term "computable." As can be seen from (2.12), (2.14), and (2.21), for each statistical model the partial derivatives $\frac{dE}{d\lambda_i}$ are of the form

$$\frac{dE}{d\lambda_i} = \int_{\hat{c}} \mathcal{R}\left(I(\mathbf{y})\right) \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} \tag{2.22}$$

with $\mathcal{R} : \mathcal{Z} \mapsto \mathbb{R}$, a function depending on the choice of statistical model.

This line integral and in particular the term $\left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle$ are difficult to compute since the parameter λ_i acts on 3D coordinates, while $\hat{c}$ and $\hat{\mathbf{n}}$ live in the 2D image plane. To facilitate computations, we now express (2.22) in the 3D world.

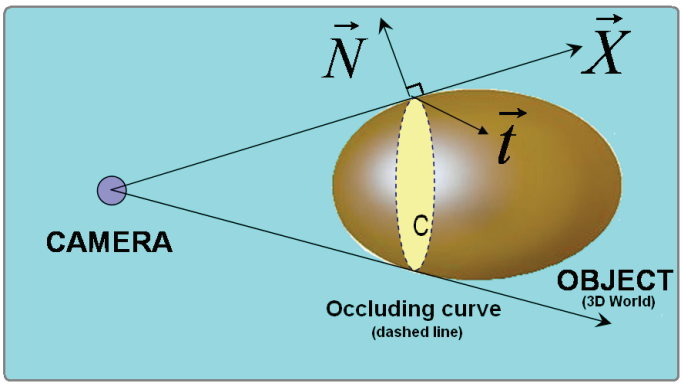


Figure 3. *Schema visualizing the occluding curve of a 3D object (dashed line) from the viewpoint of the camera and our notation in the 3D world.*

Using the arc-length s of C and the $\frac{\pi}{2}$-rotation matrix $J = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$ (ensuring that the normal vector $\hat{\mathbf{n}}$ points outwards), one has

(2.23)
$$\left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, J\frac{\partial \hat{c}}{\partial \hat{s}} \right\rangle d\hat{s} = \left\langle \frac{\partial \pi(C)}{\partial \lambda_i}, J\frac{\partial \pi(C)}{\partial s}\frac{ds}{d\hat{s}} \right\rangle d\hat{s} = \left\langle \frac{\partial \pi(C)}{\partial \lambda_i}, J\frac{\partial \pi(C)}{\partial s} \right\rangle ds.$$

Letting $\mathcal{J}$ denote the Jacobian of $\pi(\mathbf{X})$ with respect to the spatial coordinates, we have that

$$\mathcal{J} = \frac{1}{Z^2}\begin{bmatrix} Z & 0 & -X \\ 0 & Z & -Y \end{bmatrix}.$$

From (2.23), one gets

(2.24)
$$\begin{aligned} \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} &= \left\langle \mathcal{J}\frac{\partial \mathbf{X}}{\partial \lambda_i}, J\mathcal{J}\frac{\partial \mathbf{X}}{\partial s} \right\rangle ds = \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathcal{J}^T J \mathcal{J}\frac{\partial \mathbf{X}}{\partial s} \right\rangle ds \\ &= \frac{1}{Z^3}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix}\frac{\partial \mathbf{X}}{\partial s} \right\rangle ds = \frac{1}{Z^3}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} \right\rangle ds. \end{aligned}$$

In (2.24), the point $\mathbf{X}$ belongs to the occluding curve C. A necessary condition for a point $\mathbf{X}$ to belong to the occluding curve is that $\langle \mathbf{X}, \mathbf{N} \rangle = 0$ (since the associated vector $\mathbf{X}$, with origin at the center of the camera, corresponds to the projection/viewing direction and is tangent to the surface S at $\mathbf{X}$; see Figure 3). The vector $\mathbf{t} = \frac{\partial \mathbf{X}}{\partial s}$ is the tangent to the curve C at the point $\mathbf{X}$. Since the vectors $\mathbf{t}$ and $\mathbf{X}$ belong to the tangent plane to S at $\mathbf{X}$, one has $\frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} = \|\mathbf{X}\|\mathbf{N}\sin(\theta)$, with $\theta = \widehat{(\mathbf{t}, \mathbf{X})}$ the angle between $\mathbf{t}$ and $\mathbf{X}$. For $\mathbf{X} \in C$, we have that

(2.25)
$$\frac{\partial}{\partial s}\langle \mathbf{X}, \mathbf{N} \rangle = 0 = \underbrace{\left\langle \frac{\partial \mathbf{X}}{\partial s}, \mathbf{N} \right\rangle}_{=0} + \left\langle \frac{\partial \mathbf{N}}{\partial s}, \mathbf{X} \right\rangle = \langle d\mathbf{N}(\mathbf{t}), \mathbf{X} \rangle = \mathrm{II}(\mathbf{t}, \mathbf{X}).$$

Since the second fundamental form $\mathrm{II}(\mathbf{t}, \mathbf{X}) = 0$, the vectors $\mathbf{t}$ and $\mathbf{X}$ are conjugate (see [31]). Hence, using the Euler formula, one can show that $K\sin^2\theta = \kappa_X \kappa_t$, where K

is the Gaussian curvature, and κ_X and κ_t denote the normal curvatures in the directions $\mathbf{X}$ and $\mathbf{t}$ at $\mathbf{X} \in S$, respectively. Plugging this into (2.24), one gets

$$\left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_X \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds. \tag{2.26}$$

Thus, the flow becomes a simple line integral on C:

$$\frac{dE}{d\lambda_i} = \int_C \mathcal{R}\left(I\left(\pi(\mathbf{X})\right)\right) \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_X \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds. \tag{2.27}$$

We now compute the term $\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle$ when λ_i is a translation or rotation parameter:

- For $i = 1, 2, 3$ (i.e., λ_i is a translation parameter) and $\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$, one has

$$\begin{aligned} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X}_0 + \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \begin{bmatrix} \frac{\partial \lambda_1}{\partial \lambda_i} \\ \frac{\partial \lambda_2}{\partial \lambda_i} \\ \frac{\partial \lambda_3}{\partial \lambda_i} \end{bmatrix}, \mathbf{N} \right\rangle \\ = \left\langle \begin{bmatrix} \delta_{1,i} \\ \delta_{2,i} \\ \delta_{3,i} \end{bmatrix}, \mathbf{N} \right\rangle = N_i, \end{aligned} \tag{2.28}$$

  where the Kronecker symbol $\delta_{i,j}$ was used ($\delta_{i,j} = 1$ if $i = j$ and 0 otherwise).
- For $i = 4, 5, 6$ (i.e., λ_i is a rotation parameter), and using the expression of the rotation matrix written in exponential coordinates,

$$\mathbf{R} = \exp\left(\begin{bmatrix} 0 & -\lambda_6 & \lambda_5 \\ \lambda_6 & 0 & -\lambda_4 \\ -\lambda_5 & \lambda_4 & 0 \end{bmatrix} \right),$$

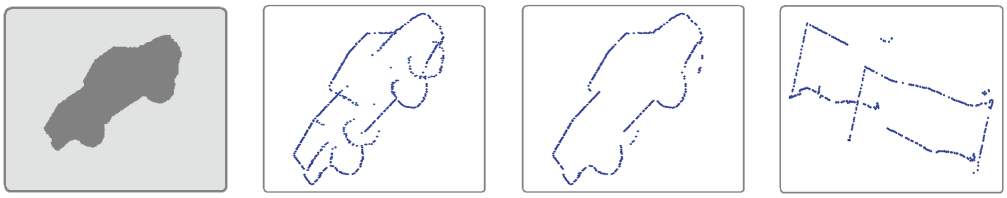
  one has

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X_0}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \mathbf{R} \begin{bmatrix} 0 & -\delta_{3,i} & \delta_{2,j} \\ \delta_{3,i} & 0 & -\delta_{1,i} \\ -\delta_{2,i} & \delta_{1,i} & 0 \end{bmatrix} \mathbf{X}_0, \mathbf{N} \right\rangle. \tag{2.29}$$

2.4. Remarks concerning the gradient term and its implementation. In (2.27), the computation of the gradients involves the explicit determination of the occluding curve C. Intuitively, this curve allows us to understand and take into account the dependency of the aspect of the object with respect to the point of view. From the definition, one can compute

$$C = \{\mathbf{X} \in \mathcal{V}^+ \cap \mathcal{V}^- \text{ such that } \pi(\mathbf{X}) \in \hat{c}\}, \tag{2.30}$$

where $\mathcal{V}^+ = \{\mathbf{X} \in S, \text{ so that (s.t.) } \langle \mathbf{X}, \mathbf{N} \rangle \geq 0\}$ and $\mathcal{V}^- = \{\mathbf{X} \in S, \text{ s.t. } \langle \mathbf{X}, \mathbf{N} \rangle \leq 0\}$.



(a) (b) (c) (d)

Figure 4. *Understanding the occluding curve.* (a) *Projection of the 3D object in the 2D image plane.* (b) *Candidates for the occluding curve (points belonging to $\mathcal{V}^+ \cap \mathcal{V}^-$) that need to be filtered with the condition "$\pi(\mathbf{X}) \in \hat{c}$."* (c)–(d) *Visualization of the occluding curve (after filtering) corresponding to the camera image on the left from different points of view in the 3D world. Note: The occluding curve is in general not a closed curve for nonconvex objects.*

In practice, the two sets $\mathcal{V}^+$ and $\mathcal{V}^-$ can be easily computed from the available data $\mathbf{X}$ and $\mathbf{N}$ and by using a small value of ϵ_1 instead of 0 in the definitions of $\mathcal{V}^+$ and $\mathcal{V}^-$ to ensure the intersection comprises a sufficient number of points:[8] $\mathcal{V}^+_{\epsilon_1} = \{\mathbf{X} \in S, \text{ s.t. } \langle \mathbf{X}, \mathbf{N} \rangle \geq -\epsilon_1\}$ and $\mathcal{V}^-_{\epsilon_1} = \{\mathbf{X} \in S, \text{ s.t. } \langle \mathbf{X}, \mathbf{N} \rangle \leq \epsilon_1\}$. In the general case of nonconvex shapes, the intersection of the two sets comprises points that project inside the 2D projection R of the 3D model (e.g., image(b) in Figure 4) and must be filtered by ensuring that the necessary and sufficient condition to belong to C, $\pi(\mathbf{X}) \in \hat{c}$, is fulfilled. This can be implemented by selecting only points such as $\|\pi(\mathbf{X}) - \hat{c}\| \leq \epsilon_2$, with ϵ_2 a chosen (small) parameter. One can obtain $\hat{c}$ by using morphological operations on R: $\hat{c} \simeq R \setminus \mathcal{E}(R)$, with $\mathcal{E}$ denoting the erosion operation for a chosen kernel [40]. Figure 4 presents different visualizations of an occluding curve computed in this fashion. One can also note that the set V (respectively, $V^c = S \setminus V$) of points $\mathbf{X} \in S$ that are visible (respectively, not visible) from the camera center is such that $V \subset \mathcal{V}^+$ (respectively, $V^c \supset \mathcal{V}^-$).

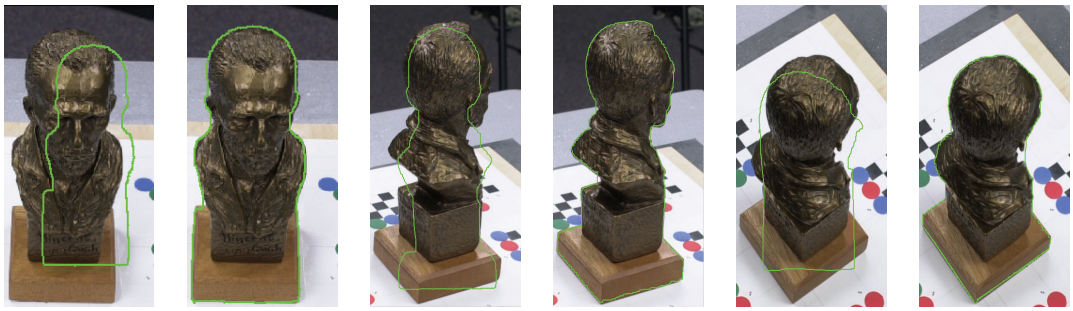
The term $\sqrt{\frac{\kappa_X \kappa_t}{K}}$ can be computed at each iteration of the algorithm using the principal curvatures and principal directions for each point $\mathbf{X} \in S$, and the Euler formula (see [31]; N.B.: the principal directions and curvatures can be precomputed). To save computational time, and noting that $\sqrt{\frac{\kappa_X \kappa_t}{K}} \geq 0$, we used the approximation $\sqrt{\frac{\kappa_X \kappa_t}{K}} \simeq 1$ in our implementation of (2.27), which still decreased the energy E. Note that this approximation is poorer when $\theta \simeq 0$. However, the condition $\theta = 0$ implies that the viewing direction $\mathbf{X}$ and the tangent to the occluding curve are identical. This occurs only for a finite number of points on the occluding curve for regular surfaces and, thus, can be expected to have little impact on the sign of the derivative $\frac{\partial E}{\partial \lambda_i}$ (which is a sum over an infinite number of points of the curve C). By contradiction, let us suppose that two neighboring points X_1 and X_2 of the occluding curve (as such X_1 and X_2 must be visible points) verify the condition $\theta = 0$ (e.g., $\theta_1 = \widehat{(\mathbf{t}, \mathbf{X_1})}$). We thus have $\mathbf{t} = \overrightarrow{X_1 X_2} = \mathbf{X}_1 = \mathbf{X}_2$, which contradicts the fact that both X_1 and X_2 are visible (either X_1 occludes X_2 or X_2 occludes X_1).

3. Experiments.

We now report experimental results obtained for both synthetic and real datasets. Different 3D models of rigid objects (see Figure 1) were used to perform segmentation

[8] We refer to the condition $\langle \mathbf{X}, \mathbf{N} \rangle = 0$, which is rarely exactly met in practice due to the sampling of the 3D surface.

Figure 5. *Robustness to initialization—segmentation of a synthetic color image. Left: initialization. Middle: intermediate steps of the evolution. Right: final result.*



(a_1) (b_1) (a_2) (b_2) (a_3) (b_3)

Figure 6. *Robustness to initialization—segmentation of natural color images.* (a_n)*'s: challenging initializations (e.g., large error in translation or angular positions (green curve)).* (b_n)*'s: final results with the proposed approach (green curve).*

and tracking tasks that highlight the robustness of our technique to *initialization*, *noise*, and *missing or imperfect information*. The shapes of the objects, notably the horse, the elephant, and the Van Gogh bust, cannot readily be described in terms of geometric primitives (lines, ellipses, etc.) or even algebraically, and thus they do not satisfy the working hypotheses of standard pose estimation techniques [2, 3, 5, 6].

3.1. Robustness to initialization. Figure 5 shows segmentation results (and 3D coordinate recoveries) obtained using our approach for a synthetic color image. Results were obtained running (2.14) until convergence. Figure 6 shows results for diverse natural color images, obtained using (2.21). Despite initializations that are quite far from the truth (e.g., large errors in translation or angular position), accurate segmentations are obtained. Figure 7 shows tracking results obtained for a real sequence, using the flow of (2.12). The sequence is composed of 32 images of a rigid toy horse. The images were taken from discrete positions of a calibrated camera that underwent a complete rotation around the object. The camera "jumps" between successive images, creating large changes in the pose of the object that needs to be recovered (e.g., changes in the angular position of the camera can exceed 15° between frames). Tracking this sequence would be challenging for many 3D pose estimation techniques available in the literature: A number of techniques using local features such as points or edges (e.g., [1, 3]) are likely to be thrown off by the textured/noisy background (false features) and get trapped in local minima. The sequence was tracked with our technique, using a very

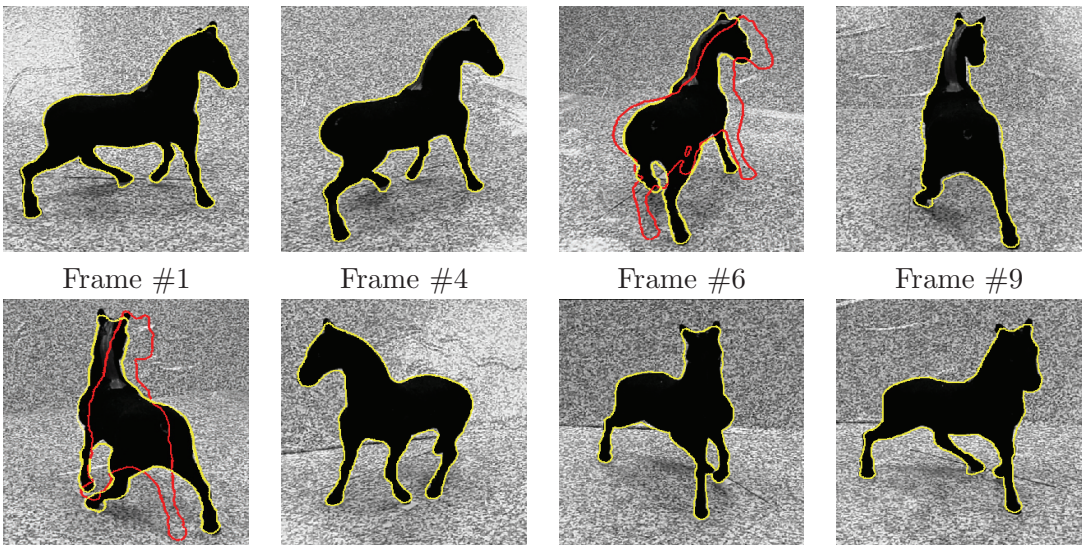


Figure 7. *Robustness to initialization—tracking a natural sequence. Yellow contours: final results after convergence. Red contours: initializations from the result of the preceding image (see text for our tracking scheme). The aspect of the object changes drastically throughout the sequence. The position of the object undergoes large changes between successive images. This sequence would pose a challenge to most 3D pose estimation algorithms that are based on local information due to the "noisy" background (false features).*

simple scheme: For each image, initialization was performed using the pose parameters corresponding to the minimum of the energy obtained for the preceding image, and our approach was run until convergence. Despite the difficulties described above, very satisfying tracking performances were observed. This highlights the robustness of the technique to initialization since the large camera jumps are accommodated and the method is not trapped in local minima. We note that to save computational time, a down-sampled and smoothed version of the 3D model obtained in [23] was used, explaining that some finer details (e.g., with high curvature such as the ears of the horse) are not captured by the segmentation. This highlights another robustness aspect of the methodology: The 3D model does not need to be perfect to lead to satisfying results. Also, it can be noticed that region-based active contour techniques, such as [10], would lead to reasonably accurate segmentations on this particular sequence. However, these approaches would not also determine the pose of the object, which is valuable information for tracking applications.

3.2. Robustness to noise. To test the robustness of our technique to noise, a sequence of 200 images was constructed by continuously transforming the 3D model of the "2D3D" logo and projecting it onto the image plane using the parameters of a simulated calibrated camera (e.g., focal length $f = 200$). The translation parameters, rotation axis, and angle were continuously varied (i.e., the total angle variation over the sequence exceeded 160°) to ensure a large variation of the aspect and position of the object throughout the sequence. From the basic sequence obtained, diverse levels of Gaussian noise were added, with standard deviation ranging from $\sigma_n = 10\%$ to $\sigma_n = 100\%$ (see Figure 8).

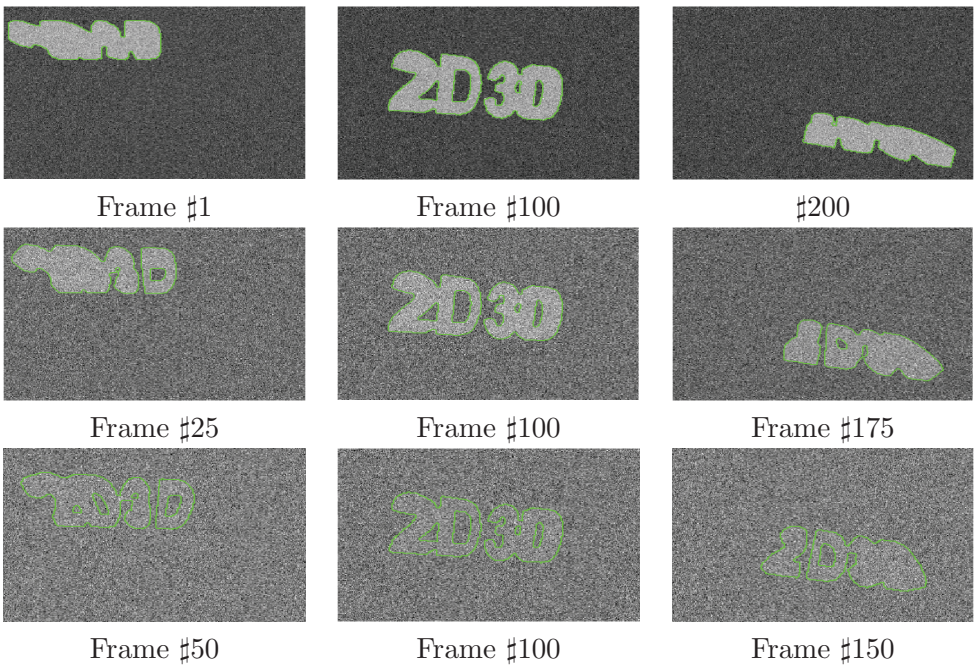


Figure 8. *Robustness to noise. Visual tracking results for the sequences involving the "2D3D" logo (green curves). First row: tracked sequence with Gaussian noise of standard deviation $\sigma_n = 10\%$. Second row: tracked sequence for $\sigma_n = 30\%$. Third row: tracked sequence for $\sigma_n = 100\%$.*

Table 1

Robustness to noise. Quantitative tracking results for the "2D3D" sequences with diverse levels of noise. The table displays %-absolute error statistics over the 200 *images of the sequences.*

Noise level	Mean error (in %)	Std. dev. error (in %)	Max error (in %)
$\sigma_n = 10\%$	**T**: 0.85; **R**: 0.96	**T**: 0.23; **R**: 0.45	**T**: 1.43; **R**: 2.60
$\sigma_n = 30\%$	**T**: 0.97; **R**: 1.09	**T**: 0.21; **R**: 0.47	**T**: 1.50; **R**: 2.94
$\sigma_n = 60\%$	**T**: 0.95; **R**: 1.30	**T**: 0.30; **R**: 0.52	**T**: 2.39; **R**: 2.60
$\sigma_n = 100\%$	**T**: 1.02; **R**: 2.12	**T**: 0.39; **R**: 0.87	**T**: 2.18; **R**: 4.36

Typical visual results obtained using our approach (flow of (2.12) combined with the tracking scheme alluded to above) are reproduced in Figure 8. For all noise levels, which can be rather large (e.g., in the case $\sigma_n = 100\%$ object and background are barely distinguishable), tracking was maintained throughout the whole sequence. Table 1 reproduces the results of the pose estimation procedure. For each image, percent *absolute* errors with respect to the ground-truth were computed for both the translation and rotation as $\text{Error} = \frac{\|\mathbf{v}_{\text{measured}} - \mathbf{v}_{\text{truth}}\|}{\|\mathbf{v}_{\text{truth}}\|}$, with $\mathbf{v}$ a translation or quaternion (see [38]) vector. From the pose estimation point of view, the method appears to perform quite well: Average error and standard deviation computed over the 200 frames of each sequence rarely exceed 2% and 1%, respectively, for *both* translation and rotation. This highlights the accuracy and reliability of the method and suggests that it is quite resilient to large amounts of noise (very little deterioration of the results is observed with increasing noise levels).

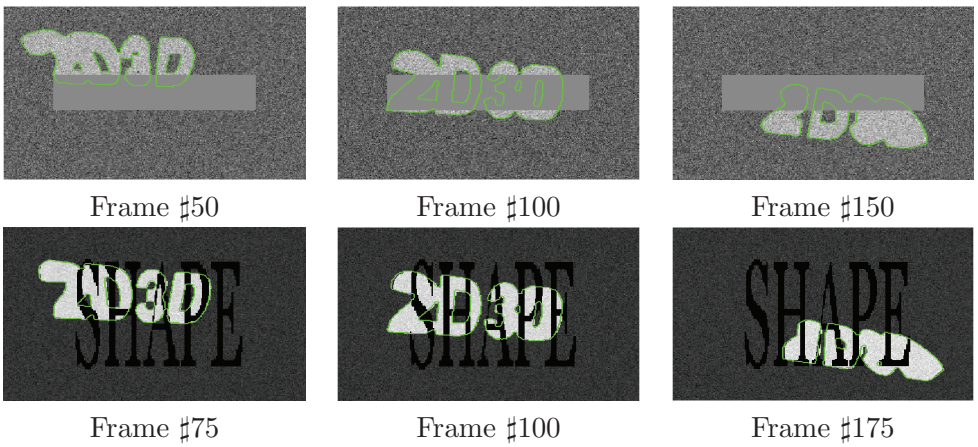


Figure 9. *Robustness to missing information. Tracking results (green curves) for the "2D3D" sequences with occlusions. First row: sequence with Rectangular occlusion. Second row: sequence with Word "SHAPE" as occlusion. Gaussian noise with* $\sigma_n = 30\%$ *was added.*

3.3. Robustness to missing/imperfect image information. To test the robustness of our technique to missing information, we created two sequences by adding two different occlusions in the basic sequence featuring the "2D3D" model (see Figure 9). The first occlusion is a gray rectangle that can mask more than 2/3 of the "2D3D" logo. The second occlusion is the word "SHAPE" written in black letters that can mask the object at several places. Gaussian noise of standard deviation 30% was also added to both resulting sequences. Figure 9 presents the results of tracking the sequences of 200 frames with our approach. One notes that despite the occlusions (and noise), accurate segmentations are obtained: In particular, missing letters or parts are accurately localized and reconstructed. Track was maintained throughout both sequences. For the first sequence mean %-absolute error (over the 200 frames) in the transformation parameters was 1.08% for translation ($\mathbf{T}$) and 1.57% for rotation ($\mathbf{R}$) with standard deviation 0.45% for $\mathbf{T}$ and 0.75% for $\mathbf{R}$. For the second sequence mean %-absolute error was 0.87% for $\mathbf{T}$ and 1.19% for $\mathbf{R}$ (standard deviation 0.34% for $\mathbf{T}$ and 0.53% for $\mathbf{R}$).

In Figure 10, we used images extracted from the horse sequence and occluded different parts of the horse body (e.g., the legs, which have valuable information about its angular position). Diverse pose parameters quite far from the truth were used as initializations (e.g., angular position could be off by more than 30°). Despite the occlusions with various pixel intensities or texture (and poor initializations), very convincing segmentations were obtained. Also, the positions of the object in the camera referential were accurately recovered. As can be noticed by comparing with Figure 7, the results in the presence of occlusion are very comparable to those without occlusion.

In Figures 11 and 12, we present segmentation results where the background and object are difficult or impossible to distinguish based on pixel statistics only (due to specular reflections on the object, similar colors in object and background, or occlusions). The results obtained with the (infinite dimensional) active contour flow of [11], which is the region-based segmentation technique underlying our approach, are not satisfying since the contour leaks into the background. Robust results are obtained using our approach.

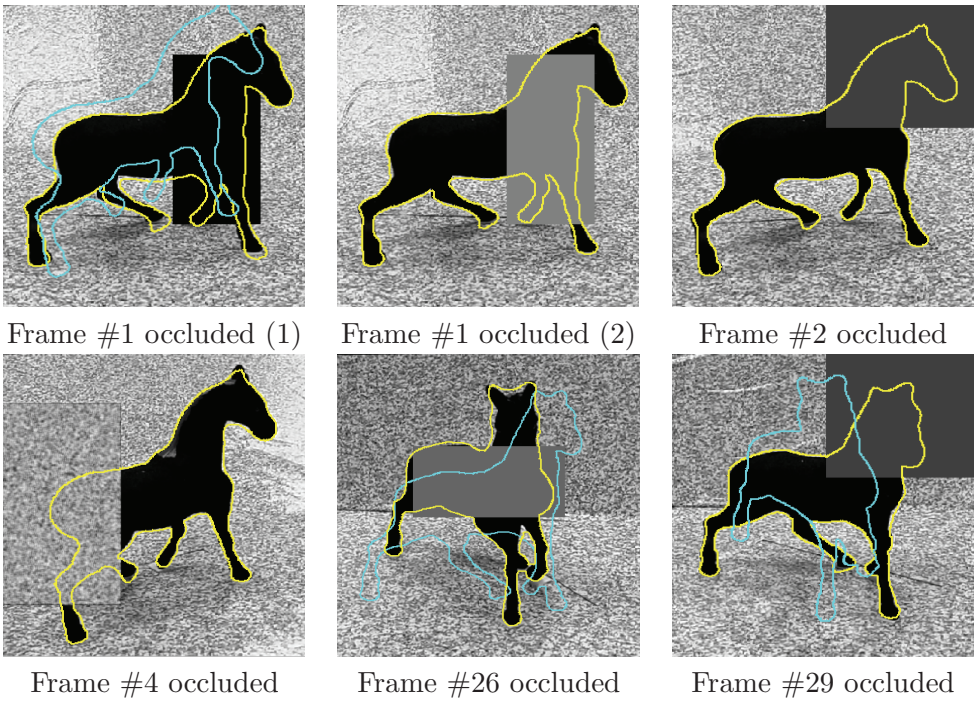


Figure 10. *Robustness to missing information. Segmentation results with occlusions. Cyan contours: some of the initializations tested (note the large errors in angular position). Yellow contours: final results (almost identical to results in Figure 7 with no occlusion).*

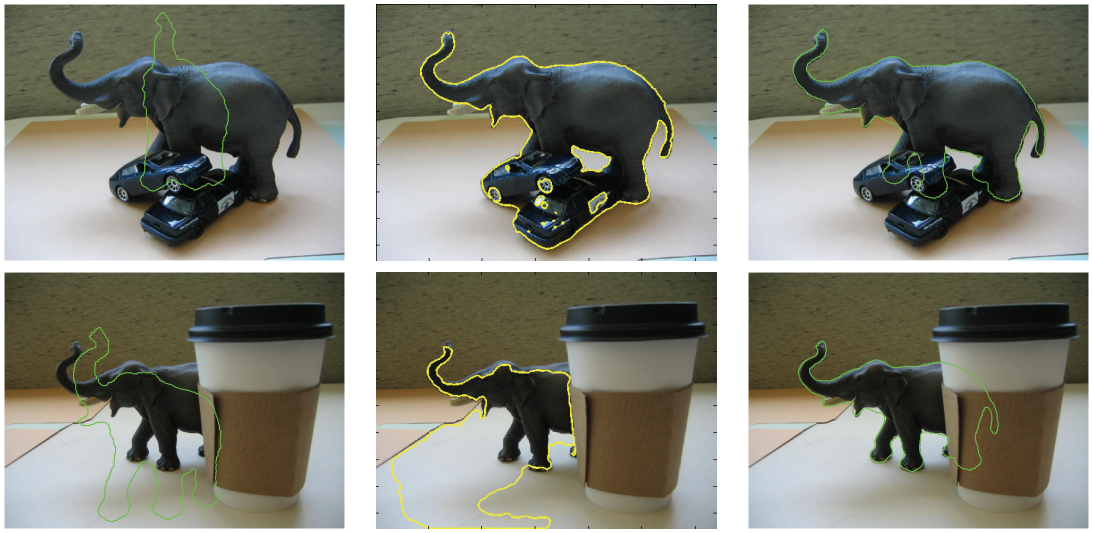


Figure 11. *Robustness to imperfect or missing information. Comparative segmentation results with occlusions. Left: initializations (green curves). Middle: final results obtained with (infinite dimensional) active contour flow as in [11], which is the region-based segmentation technique underlying our approach (yellow curves). Right: final results with our approach (green curves). In these images, statistical distinction between object and background is difficult due to similar colors in object and background and occlusion of parts of the object.*

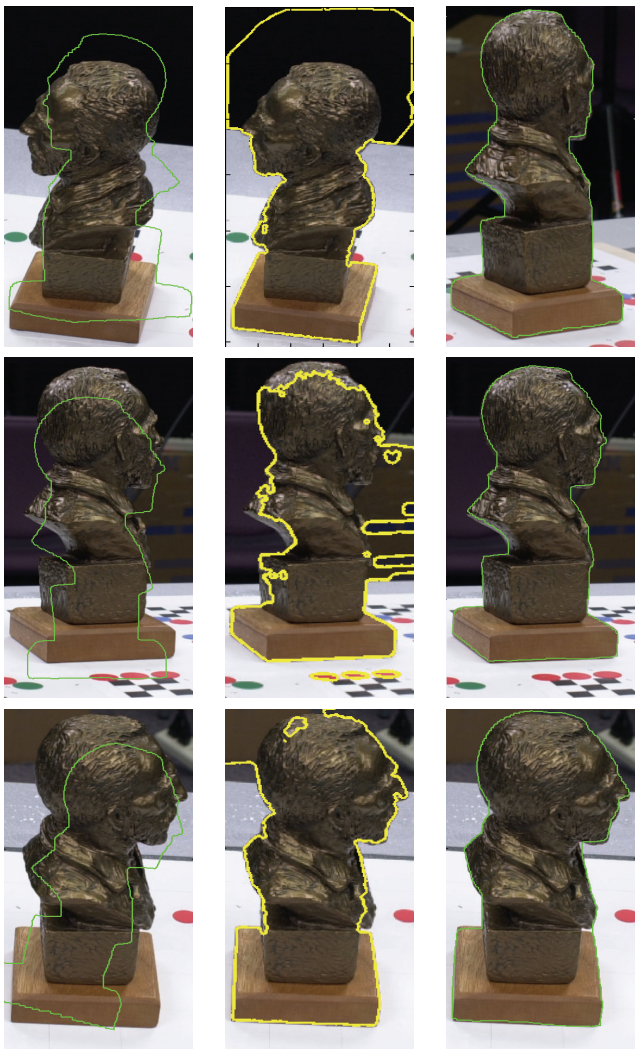
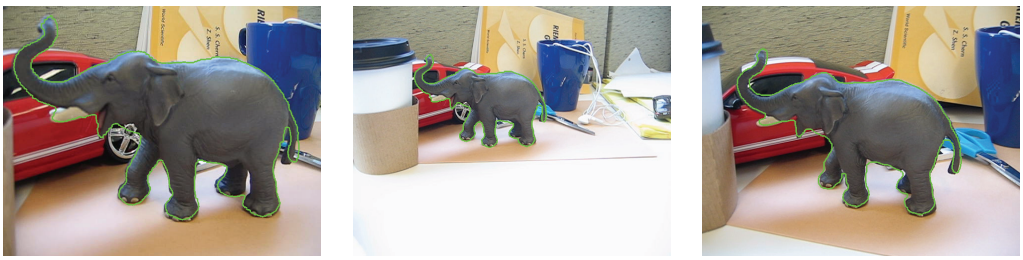


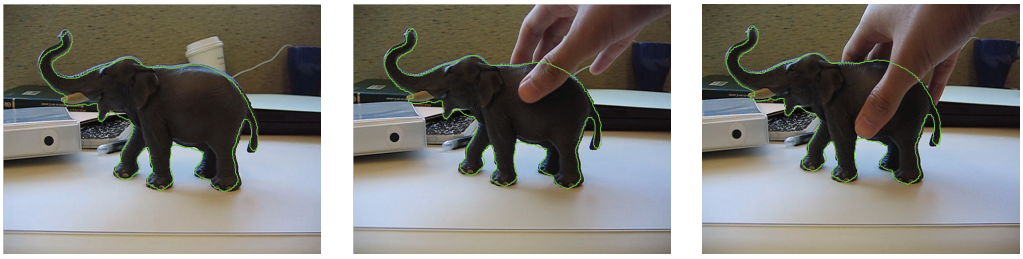
Figure 12. *Robustness to imperfect information. Comparative segmentation results. Left: initializations (green curves). Middle: final results obtained with (infinite dimensional) active contour flow as in* [11]*, which is the region-based segmentation technique underlying our approach (yellow curves). Right: final results with our approach (green curves). In these images, statistical distinction between object and background is difficult due to specularities on the object and similar colors in object and background.*

The experiments of Figures 9, 10, 11, and 12 would pose a major challenge to most region-based active contour techniques, even using shape priors [17, 19, 20]: Statistics alone are not sufficient to segment the images, and the aspect of the object changes drastically from one image to the other. Hence, a large catalogue of 2D shapes would need to be learned to achieve similar performances using the method in [17, 19, 20], for instance.

3.4. Tracking sequences. In this section, we present tracking results for three challenging sequences of images. The first two sequences are composed of 250 frames. In addition to a cluttered background, important changes in the size and aspect of the object occur due to


Seq. #1 ; Frame #1 Seq. #1 ; Frame #100 Seq. #1 ; Frame #250


Seq. #2 ; Frame #1 Seq. #2 ; Frame #100 Seq. #2 ; Frame #250


Seq. #3 ; Frame #1 Seq. #3 ; Frame #40 Seq. #3 ; Frame #50

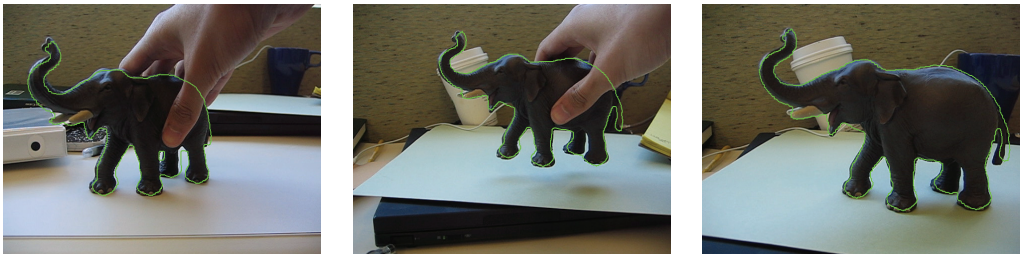

Seq. #3 ; Frame #100 Seq. #3 ; Frame #300 Seq. #3 ; Frame #450

Figure 13. *Tracking results for 3 sequences (green curves). Note in particular the cluttered background, partial occlusions, and fast changes in scale.*

camera motion. The third sequence is composed of 450 frames. In this sequence, the object is manually moved, which creates a partial occlusion as well as changes in the background and angular position of the object. Using the flow of (2.21) and our tracking scheme, the three sequences were convincingly tracked in their integrality. Figure 13 presents some of the typical results obtained.

4. Conclusions and future work. In this work, we presented a region-based approach to the 3D pose estimation problem. This approach differs from other 3D pose estimation

algorithms since it does not rely on local image features. Our method allows one to employ global image statistics to drive the pose estimation process. This confers a satisfying level of robustness to noise and initialization to our framework and bypasses the need to establish correspondences between image and object features, contrary to most 3D pose estimation approaches.

Furthermore, the approach possesses the typical qualities of a region-based active contour technique with shape prior, such as robustness to occlusion or missing information, without the need to evolve an infinite dimensional contour. Also, the prior knowledge of the shape of the object is compactly represented by a unique 3D model, instead of a dense catalogue of 2D shapes.

The main advantage of the proposed technique is that it enables one to locate the object not only in 2D images (a task typically handled by GAC approaches) but also in the world (a task typically handled by 2D-3D pose estimation algorithms). This makes the method particularly suitable for tracking applications involving a unique calibrated camera.

A possible direction for future research is to extend the proposed approach to include the knowledge of multiple 3D shapes. In particular, the method in [18] (where evolution of parameters in the shape space is performed in addition to pose parameters) could be adapted to the problem at hand. It is expected that the resulting framework will allow one to learn the possible deformations of the object and lead to robust performances for nonrigid registration and tracking tasks.

REFERENCES

[1] L. Quan and Z.-D. Lan, *Linear n-point camera pose determination*, IEEE Trans. Pattern Anal. Mach. Intell., 21 (1999), pp. 774–780.

[2] M. Dhome, M. Richetin, and J.-T. Lapreste, *Determination of the attitude of 3D objects from a single perspective view*, IEEE Trans. Pattern Anal. Mach. Intell., 11 (1989), pp. 1265–1278.

[3] E. Marchand, P. Bouthemy, and F. Chaumette, *A 2D-3D model-based approach to real-time visual tracking*, Image Vision Comput., 19 (2001), pp. 941–955.

[4] M. Zerroug and R. Nevatia, *Pose estimation of multi-part curved objects*, in Proceedings of the International Symposium on Computer Vision (ISCV '95), 1995, p. 431.

[5] B. Rosenhahn, C. Perwass, and G. Sommer, *Pose estimation of free-form contours*, Int. J. Comput. Vision, 62 (2005), pp. 267–289.

[6] T. Drummond and R. Cipolla, *Real-time tracking of multiple articulated structures in multiple views*, in Proceedings of the 6th European Conference on Computer Vision (ECCV), 2000, pp. 20–36.

[7] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, Int. J. Comput. Vision, 22 (1997), pp. 61–79.

[8] S. Kichenassamy, S. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, *Conformal curvature flow: From phase transitions to active vision*, Arch. Rational Mech. Anal., 134 (1996), pp. 275–301.

[9] S. Chun Zhu and A. L. Yuille, *Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., 18 (1996), pp. 884–900.

[10] T. Chan and L. Vese, *Active contours without edges*, IEEE Trans. Image Process., 10 (2001), pp. 266–277.

[11] N. Paragios and R. Deriche, *Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision*, J. Vis. Commun. Image Represent., 13 (2002), pp. 249–268.

[12] S. Dambreville, A. Yezzi, M. Niethammer, and A. Tannenbaum, *A variational framework combining level-sets and thresholding*, in proceedings of the British Machine Vision Conference (BMVC), 2007, pp. 266–280.

[13] O. Michailovich, Y. Rathi, and A. Tannenbaum, *Image segmentation using active contours driven by the Bhattacharyya gradient flow*, IEEE Trans. Image Process., 16 (2007), pp. 2787–2801.

[14] J. Kim, J. Fisher, A. Yezzi, M. Cetin, and A. Willsky, *Nonparametric methods for image segmentation using information theory and curve evolution*, in Proceedings of the 2002 IEEE International Conference on Image Processing (ICIP), Vol. 3, 2002, pp. 797–800.

[15] M. Rousson and D. Cremers, *Efficient kernel density estimation of shape and intensity priors for level set segmentation*, in Procceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Vol. 2, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 757–764.

[16] N. Paragios and R. Deriche, *Geodesic active regions for supervised texture segmentation*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Vol. 2, 1999, pp. 926–932.

[17] M. Leventon, E. Grimson, and O. Faugeras, *Statistical shape influence in geodesic active contours*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2000, pp. 316–323.

[18] A. Tsai, T. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, E. Grimson, and A. Willsky, *A shape-based approach to the segmentation of medical imagery using level sets*, IEEE Trans. Med. Imaging, 22 (2003), pp. 137–153.

[19] D. Cremers, T. Kohlberger, and C. Schnoerr, *Shape statistics in kernel space for variational image segmentation*, Pattern Recognition, 36 (2003), pp. 1929–1943.

[20] S. Dambreville, Y. Rathi, and A. Tannenbaum, *Shape-based approach to robust image segmentation using kernel PCA*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 977–984.

[21] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New York, 2003.

[22] G. Turk and M. Levoy, *Zippered polygon meshes from range images*, in Proceedings of SIGGRAPH, ACM, New York, 1994, pp. 311–318.

[23] A. Yezzi and S. Soatto, *Structure from motion for scenes without features*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2003, pp. 171–178.

[24] S. Dambreville, R. Sandhu, A. Yezzi, and A. Tannenbaum, *Robust 3D pose estimation and efficient 2D region-based segmentation from a 3D shape prior*, in Proceedings of the European Conference on Computer Vision (ECCV), 2008, pp. 169–182.

[25] O. D. Faugeras and R. Keriven, *Variational principles, surface evolution PDEs, level set methods, and the stereo problem*, IEEE Trans. Image Process., 7 (1998), pp. 336–344.

[26] A. Yezzi and S. Soatto, *Stereoscopic segmentation*, Int. J. Comput. Vision, 53 (2003), pp. 31–43.

[27] G. Unal, A. Yezzi, S. Soatto, and G. Slabaugh, *A variational approach to problems in calibration of multiple cameras*, IEEE Trans. Pattern Anal. Mach. Intell., 29 (2007), pp. 1322–1338.

[28] T. Riklin-Raviv, N. Kiryati, and N. Sochen, *Exploiting occluding contours for real-time 3D tracking: A unified approach*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Vol. 1, 2005, pp. 204–211.

[29] B. Rosenhahn, T. Brox, and J. Weickert, *Three-dimensional shape knowledge for joint image segmentation and pose tracking*, Int. J. Comput. Vision, 73 (2007), pp. 243–262.

[30] C. Schmaltz, B. Rosenhahn, T. Brox, D. Cremers, J. Weickert, L. Wietzke, and G. Sommer, *Region-based pose tracking*, in Pattern Recognition and Image Analysis, Lecture Notes in Comput. Sci. 4478, Springer-Verlag, Berlin, 2007, pp. 56–63.

[31] M. P. DoCarmo, *Differential Geometry of Curves and Surfaces*, Prentice–Hall, Englewood Cliffs, NJ, 1976.

[32] M. Bray, P. Kohli, and P. Torr, *Posecut: Simultaneous segmentation and 3D pose estimation of humans using dynamic graph-cuts*, in Procceedings of the European Conference on Computer Vision (ECCV), Vol. 2, 2006, pp. 642–655.

[33] P. Kohli, J. Rihan, M. Bray, and P. Torr, *Simultaneous segmentation and pose estimation of humans using dynamic graph cuts*, Int. J. Comput. Vision, 79 (2008), pp. 285–298.

[34] V. Lepetit and P. Fua, *Monocular model-based 3D tracking of rigid objects: A survey*, Found. Trends Comput. Graph. Vis., 1 (2005), pp. 1–89.

[35] G. Li, Y. Tsin, and Y. Genc, *Exploiting occluding contours for real-time 3D tracking: A unified approach*, in Proceedings of the IEEE Conference on Computer Vision (ICCV), 2007, pp. 1–8.

[36] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice–Hall, Englewood Cliffs, NJ, 2003.
[37] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.
[38] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3D Vision*, Springer-Verlag, New York, 2005.
[39] M. Rousson and R. Deriche, *A variational framework for active and adaptative segmentation of vector valued images*, in Proceedings of the Workshop on Motion and Video Computing, IEEE Computer Society, Washington, DC, 2002.
[40] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed., Prentice–Hall, Upper Saddle River, NJ, 2008.