

Polygonal and Polyhedral Contour Reconstruction in Computed Tomography

Charles Soussen and Ali Mohammad-Djafari

© 2004 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Abstract—This paper is about three-dimensional (3-D) reconstruction of a binary image from its X-ray tomographic data. We study the special case of a *compact* uniform polyhedron totally included in a uniform background and directly perform the polyhedral surface estimation. We formulate this problem as a nonlinear inverse problem using the Bayesian framework. Vertice estimation is done without using a voxel approximation of the 3-D image. It is based on the construction and optimization of a regularized criterion that accounts for surface smoothness. We investigate original deterministic local algorithms, based on the exact computation of the line projections, their update, and their derivatives with respect to the vertice coordinates. Results are first derived in the two-dimensional (2-D) case, which consists of reconstructing a 2-D object of deformable polygonal contour from its tomographic data. Then, we investigate the 3-D extension that requires technical adaptations. Simulation results illustrate the performance of polygonal and polyhedral reconstruction algorithms in terms of quality and computation time.

Index Terms—Block relaxation algorithms, closed contour and surface reconstruction, deformable template, global and local iterative descent algorithms, polygonal and polyhedral modeling, two-dimensional and three-dimensional binary image reconstruction from projections, X-ray tomography.

I. INTRODUCTION

IN THE LAST decade, image reconstruction from X-ray data has been intensively studied to produce high-resolution two-dimensional (2-D) and three-dimensional (3-D) maps of densities. However, 3-D reconstruction methods often involve a large number of parameters, thus implying a significant time of computation. In many applications, such as medical imaging or nondestructive evaluation (NDE) of materials, the reconstructed image is utilized to compute the characteristics (position, volume, and shape) of an intersected object (organ or air bubble inside a metal) by the means of a further image transformation, for example, an image segmentation task. For those issues, it is of great interest to directly model the only *contour* of the shapes of interest, hence significantly decreasing both number of parameters and computation burden. In this

paper, we consider the reconstruction of a 3-D binary image composed of a uniform compact object totally included in a uniform background. By *compact object*, we refer to a closed volume of uniform density that does not contain inclusions. The 3-D map of density $f(x, y, z)$ is then characterized by the closed surface C^* embedding the object: $f(x, y, z) = 1$ if (x, y, z) lays inside C^* ; 0, otherwise. 3-D reconstruction is then equivalent to the estimation of C^* from the X-ray data. Although compact object images are particular, their reconstruction is a first step toward that of more general images, parameterized by their local attributes. Moreover, NDE applications involve and aim to reconstruct such images, which represent a fault, for example, an air occlusion included in a homogeneous metallic material [1], [2].

Classic methods to reconstruct a 3-D map of densities from its projections are based on the volume discretization into a set of voxels. The reconstruction is then equivalent to the estimation of the density of all the voxels [3], [4]. Those methods are fairly simple due to the linearity of the projection operator. However, they appear consuming, both in memory storage and computation burden for high resolution images. On the other hand, the direct use of deformable contours in image reconstruction has been very limited because of the nonlinearity of the direct model, relating the surface parameters and the projection values. For those applications, explicit contour models are mostly parametric, and often involve a very few parameters [5]–[8].

More generally, closed deformable contours have been mainly used in the literature of computer vision and image segmentation. The former applications involve contour and surface reconstruction from silhouettes, giving rise to specific surface modeling [9], [10]. In both application fields, explicit contours, defined by their position vector, are either directly described by their arc lengths [11], [12] or by a deformable (parametric) function of a reference contour to define a class of homogeneous shapes sharing common topologic characteristics [13]. In image segmentation, nonparametric deformable templates have received a considerable attention, either in their explicit or implicit form. The former mainly refer to active contours, or snakes, defined as continuous functions of their arc length(s) [12], whereas the latter involve level-set contours of great use to efficiently cope with changes of topology [14].

The problem of shape reconstruction from X-ray projections was first encountered in medical imaging applications. The pioneering contributions involve very simple shapes, as a set of 2-D parallel ellipses for 3-D vascular tree modeling in angiographic images [5]–[7]. Handling an image formed of elliptic

Manuscript received March 14, 2003; revised January 22, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark R. Luetzgen.

C. Soussen was with the Laboratoire des Signaux et Systèmes, Centre National de la Recherche Scientifique (CNRS), Supélec, 91192 Gif-sur-Yvette Cedex, France. He now is with the Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI-CNRS), F-91403 Orsay Cedex, France (e-mail: charles.soussen@lmsi.fr).

A. Mohammad-Djafari is with the Laboratoire des Signaux et Systèmes, Centre National de la Recherche Scientifique (CNRS), Supélec, 91192 Gif-sur-Yvette Cedex, France (e-mail: djafari@lss.supelec.fr).

Digital Object Identifier 10.1109/TIP.2004.836159

objects is a simple task, since computation of its 2-D projections is direct and analytic [15]. In emission tomography, simple parametric shapes were also studied for brain tumor reconstruction using an affine transformation of a reference object of elliptic shape [16]. For the aforementioned models, the shape parameters correspond to the geometric characteristics of the ellipses (center and moments of inertia) and the affine transformation parameters, respectively. More general shape models mainly include 2-D deformable spline templates parameterized by the position of their control points. This modeling engenders *local* contours in the sense that each parameter only controls a local part of the surface [17]–[19]. In particular, first-order splines, i.e., polygonal shapes parameterized by their vertice positions, are of interest since their reconstruction from noiseless projections is exact [20]. However, the reconstruction algorithm, which requires estimation of the geometric moments of the polygonal image, is not of great use for noisy data, and was only performed on polygons with less than five vertices. Higher order spline contours, as cubic splines, have been considered for efficiently modeling 2-D shapes that are differentiable with respect to (w.r.t.) their arc lengths [17]. However, their reconstruction as well as those of polyhedral and cubic spline surfaces [21], [22] require implicit voxel representation for technical purposes. Among nonparametric shape models, the level-set approach has been investigated in the 2-D case but suffers from a huge computation cost due to the storage and update of a 3-D image [14]. This technique is however very attractive to simultaneously model multiple closed contours, whose number is not known a priori [23], [24].

In this paper, we focus on a parametric explicit representation of the contour with no need of voxel modeling. Among all finite parameterizations, we consider that an efficient model is a good compromise between the number of parameters and the variety of the generated surfaces. In particular, local models, like spline contours, are well adapted to generate heterogeneous surfaces. Moreover, the choice of model is subject to the following technical difficulties:

- 1) the direct and exact calculation of the projections of the binary image and of their derivative w.r.t. the parameters;
- 2) the verification, with a low computation cost, that the surface does not self intersect.

The second condition (nondegeneracy) is not specific to the reconstruction problem but inherent in dealing with deformable shapes. The nonlinearity of the shape projection operator and the difficulty to satisfy both conditions justify the lack of contributions in contour reconstruction from computed tomography data, especially in the 3-D case.

Specific models, as spherical harmonics, can lead to non-convex star-shaped surfaces [25], [26]. Consequently, they structurally fulfill the nondegeneracy constraint. However, those models are not local w.r.t. their parameters, and, hence, hardly generate surfaces with local irregularities. Within local models, we focus on low-order spline surfaces, which enable the modeling of complex shapes with spatial inhomogeneities. If efficient verification of the second property raises technical difficulties [17], [21], low-order splines allow the exact calculation of the contour projection. In particular, first-order splines, i.e., 2-D polygons and 3-D polyhedra, lead

to very simple projection algorithms with no use of a voxel parameterization [27]. Indeed, projection values only depend on the intersections of the projection rays with the polygon or polyhedron faces.

The rest of the paper is dedicated to first-order spline surface reconstruction. We first develop *polygonal* reconstruction schemes in the 2-D case, and then extend those results to polyhedral reconstruction. Both cases share similarities, although the latter raises additional technical difficulties. Section II introduces the reconstruction problem, which consists of the direct estimation of the contour parameters from the computed tomography data. In Section III, we investigate polygonal contour reconstruction, and perform the estimation of vertices in the Bayesian framework, leading to the minimization of a $2n$ -dimensional functional, where n is the number of vertices. We implement original local optimization algorithms (gradient and block coordinate descent), which require the update of the projection values and the exact calculation of their derivatives w.r.t. the vertice positions. In Section IV, we examine the polyhedron case and still put the stress on the algorithmic aspects of the vertice estimation. We design first-order descent algorithms that satisfy properties 1) and 2), and require technical adaptations of the 2-D procedures. Finally, Section V illustrates the polygon and polyhedron reconstruction methods on a set of synthetic limited angle data. Simulations afford comparisons between the gradient and the block coordinate-based algorithms in terms of quality and computation burden.

II. PROBLEM STATEMENT

In this section, we formulate the reconstruction of a polyhedral contour from its X-ray projections. We first define the binary parametric surface modeling and the X-ray projection operator, and then state the parameter estimation problem. The 2-D case, which is very similar, will be briefly formulated at the beginning of Section III.

A. Closed-Surface Modeling

Considering the uniformity and compacity assumptions, the density function $f(x, y, z)$ is binary, and only characterized by the object closed surface \mathcal{C}^*

$$f(x, y, z) = \begin{cases} a, & \text{if } (x, y, z) \text{ lays inside } \mathcal{C}^* \\ b, & \text{otherwise} \end{cases} \quad (1)$$

where a and b are two known gray levels, respectively set to 1 and 0, without loss of generality. To reconstruct the 3-D image $f(x, y, z)$, we model surface \mathcal{C}^* by its approximate parametric form \mathcal{C}_v . Then, reconstruction identifies with the estimation of parameters v . For simplicity, we will substitute \mathcal{C}_v by v when no ambiguity is possible.

In the case of a polyhedron of fixed faces, the parameters are the coordinates of the vertices $v_i \in \mathbb{R}^3$. Up to the ordering of vertices, we define the parameter vector $v \in \mathbb{R}^{3n}$ by appending all vertice positions: $v = [v'_1, v'_2, \dots, v'_n]'$, where $'$ denotes vector transposition. This model and the subset of \mathbb{R}^{3n} formed of unintersected closed polyhedra will be precisely defined in Section IV.

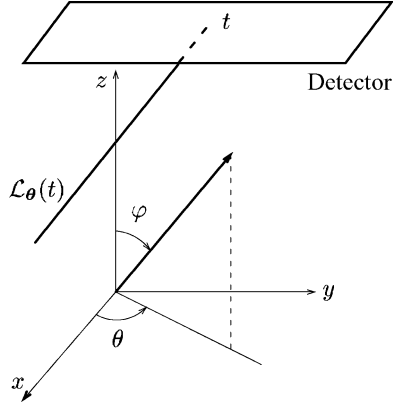


Fig. 1. Definition of spherical coordinates $\theta = [\theta, \varphi]' \in [0, 2\pi) \times [0, \pi]$ for projection line parameterization. All projection measurements are done on a single horizontal plane. The k th projection image is yielded by projection values along $\mathcal{L}_{\theta_k}(t)$ for discrete values of $t \in \mathbb{R}^2$.

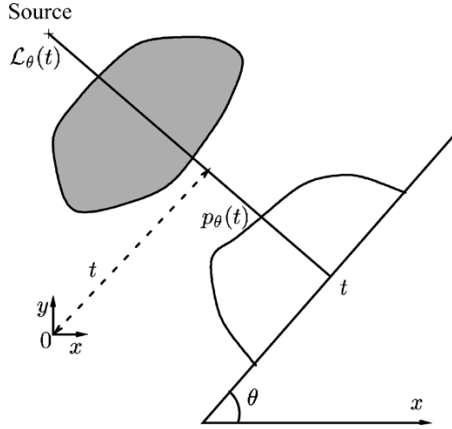


Fig. 2. Two-dimensional line projection parameterization. Projection lines $\mathcal{L}_{\theta}(t)$ are described by their angle $\theta \in [0, 2\pi)$ and detector position $t \in \mathbb{R}_+$. In the case of a binary compact object, a projection value $p_{\theta}(t)$ is equal to the length of the intersection between the object volume and projection ray $\mathcal{L}_{\theta}(t)$.

B. X-Ray Projection Operator

X-ray projections of the 3-D image are defined as its line integrals. Denoting $\mathcal{L}_{\theta}(t)$ the line oriented by spherical angle θ and passing through the detector pixel of center t (see Fig. 1), the projection along that ray is modeled by

$$p_{\theta}(t) = \int_{\mathcal{L}_{\theta}(t)} f(x, y, z) dl. \quad (2)$$

For the density function (1) yielded by surface C^* , a projection corresponds to the length of the intersection between the object volume and the projection ray: $p_{\theta}(t) = \int_{\mathcal{L}_{\theta}(t) \cap C^*} dl$. See Fig. 2 for illustration.

We now form the parallel projection of the image $f(x, y, z)$ at a fixed direction $\theta = \theta_k (k = 1, \dots, m)$. It is yielded by the set of discrete values $p_{\theta_k}(t_{ij}^k)$, $i = 1, \dots, I$, $j = 1, \dots, J$ on a parallel grid of the detector plane, and represented by one-dimensional vector \mathbf{p}_k of size IJ .¹ Then, the “global” projection vector \mathbf{p} , corresponding to all m angles θ_k , is the result of the concatenation of those vectors: $\mathbf{p} = [\mathbf{p}'_1, \dots, \mathbf{p}'_m]'$. The surface

¹The choice of parallel projection is done for the sake of simplicity. All algorithms stated below are extendable to the cone beam projection modality.

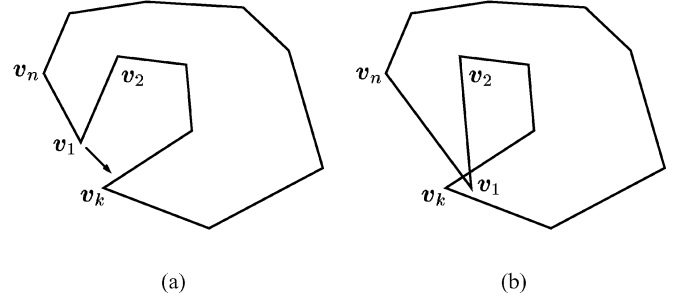


Fig. 3. Acceptability condition for polygons. (a) Acceptable polygon, (b) non-acceptable polygon yielded by (a), and a displacement of vertex \mathbf{v}_1 .

projection operators are finally defined by $C \rightarrow \mathcal{A}(C) = \mathbf{p}$ and $\mathbf{v} \in \mathbb{R}^{3n} \rightarrow \mathcal{A}(\mathbf{v}) \equiv \mathbf{p} = \mathcal{A}(C_{\mathbf{v}})$.

C. Parameter Estimation From Noisy Projections

In both the fields of medical radiography imaging and nondestructive evaluation, the line projection modeling is often justified by the high resolution of the detector pixels. The data then correspond to the noisy projections of the sought image, where the noise accounts for both errors of projection modeling and measurement. Data formation finally writes

$$\mathbf{d} = \mathcal{A}(\mathbf{v}) + \mathbf{n} \quad (3)$$

where \mathbf{n} is the noise vector, assumed to be additive, independent identically distributed (i.i.d.), white and Gaussian for the sake of simplicity. For compact object reconstruction, we also need to assume that the set of the radiograph positions t_{ij}^k totally includes the object projection supports.

Given the above modeling and assumptions, the surface reconstruction problem becomes the estimation of \mathbf{v} from the data \mathbf{d} , which can be done by any classic statistical estimation technique, e.g., maximum likelihood (ML) or Bayesian maximum a posteriori (MAP).

III. POLYGONAL SHAPE RECONSTRUCTION

A. Acceptable Polygon Modeling

Polygonal reconstruction from 2-D projections is very similar to the polyhedron reconstruction problem stated above. We consider a *direct* polygon $C_{\mathbf{v}} \equiv \mathbf{v} = [\mathbf{v}'_1, \dots, \mathbf{v}'_n]'$ $\in \mathbb{R}^{2n}$, where $\mathbf{v}_i \in \mathbb{R}^2$ denote the position of the vertices, indexed in the counterclockwise order. Two-dimensional line projections write

$$p_{\theta}(t) = \int_{\mathcal{L}_{\theta}(t)} f(x, y) dl = \int_{\mathcal{L}_{\theta}(t) \cap C_{\mathbf{v}}} dl \quad (4)$$

where the circular projection parameters $(\theta, t) \in [0, 2\pi) \times \mathbb{R}_+$ are illustrated in Fig. 2.

We define the set \mathbb{A}_n of direct polygons that do not self-intersect (*acceptable* polygons) by

$$\mathbb{A}_n = \{ \mathbf{v} \in \mathbb{R}^{2n}, \mathbf{v} \text{ is direct and } \forall i, j \in \{1, \dots, n\} \\ 1 < |i - j| < n - 1 \Rightarrow [\mathbf{v}_i, \mathbf{v}_{i+1}] \cap [\mathbf{v}_j, \mathbf{v}_{j+1}] = \emptyset \} \quad (5)$$

where, by circular extension, we denote $\mathbf{v}_{n+1} \equiv \mathbf{v}_1$. The acceptability condition is illustrated in Fig. 3.

Remark 1: The polygon parameterization is invariant by any circular permutation of the vertices. Consider a polygon $\mathbf{v} = [\mathbf{v}'_1, \dots, \mathbf{v}'_n]' \in \mathbb{A}_n$, and a circular permutation of its vertices, e.g., vector $\mathbf{w} = [\mathbf{v}'_2, \mathbf{v}'_3, \dots, \mathbf{v}'_n, \mathbf{v}'_1]'$. Clearly, \mathbf{v} and \mathbf{w} represent the same polygon, whose vertices are ordered in a different fashion. In consequence, there are n distinct parameterizations of a closed direct polygon.

B. Direct Statistical Estimation

We now focus on polygons with a large number of vertices. This number being fixed, reconstruction is done by the direct estimation of vertices $\hat{\mathbf{v}} \in \mathbb{A}_n$ from the tomographic data by the means of statistical techniques [28], [29].

1) *Maximum Likelihood:* First, ML estimation was used to reconstruct a polygon with a very low number of vertices ($n \leq 10$) [8]. Considering the assumptions on the errors of the data model $\mathbf{d} = \mathcal{A}(\mathbf{v}) + \mathbf{n}$, the ML estimator is equal to the least-square estimator

$$\begin{aligned} \hat{\mathbf{v}}_{\text{ml}} &= \arg \min_{\mathbf{v} \in \mathbb{A}_n} \{-\log \text{Pr}(\mathbf{d}|\mathbf{v})\} \\ &= \arg \min_{\mathbf{v} \in \mathbb{A}_n} \left\{ \|\mathbf{d} - \mathcal{A}(\mathbf{v})\|^2 = \sum_{i,k} [d_i^k - p_{\theta_k}(t_i^k)]^2 \right\} \quad (6) \end{aligned}$$

where d_i^k denotes the i th data value for projection k . A minimum description length (MDL) approach was also studied to estimate a polygon whose number of vertices is variable [8].

Remark 2: Criterion $\mathbf{v} \rightarrow \|\mathbf{d} - \mathcal{A}(\mathbf{v})\|^2$ is multimodal, hence nonconvex. More precisely, there are at least n global minimizers of $\|\mathbf{d} - \mathcal{A}(\mathbf{v})\|^2$, since polygon $\hat{\mathbf{v}}_{\text{ml}}$ can be parameterized by n distinct vectors.

Remark 3: Operator \mathcal{A} is not differentiable at some particular polygons, for which left and right derivatives are however defined. See Appendix I for the case of a triangle.

2) *Maximum a posteriori:* Setting aside the difficulty of optimization related to the above remarks, the accuracy of the ML estimator largely depends on the number of parameters and the number and angles of projection. It is well known that general image reconstruction from incomplete projections is an ill-posed inverse problem, as well as the reconstruction of a polygon with a large number of vertices [8], [28]. Regularization is then relevant to provide a stable solution w.r.t. the data. The MAP estimator is defined by

$$\hat{\mathbf{v}} \equiv \hat{\mathbf{v}}_{\text{map}} = \arg \min_{\mathbf{v} \in \mathbb{A}_n} \left\{ \mathcal{J}(\mathbf{v}) = \|\mathbf{d} - \mathcal{A}(\mathbf{v})\|^2 + \lambda \mathcal{R}(\mathbf{v}) \right\} \quad (7)$$

where hyperparameter $\lambda > 0$ controls the tradeoff between the fidelity of the model to the data and the shape regularization term $\mathcal{R}(\mathbf{v})$. As in image segmentation applications, we refer to the latter term as prior information on the spatial smoothness of the polygon.

For continuous active contours, smoothness basically involves the curvature, yielded by the second-order derivative of the contour w.r.t. its arc length [12], [30]. Because polygons are not twice differentiable shapes, we need to define an approximate curvature at a vertice \mathbf{v}_i , based on either the distance from

\mathbf{v}_i to the middle of its neighbors or the angle at \mathbf{v}_i , or on both criteria [28], [31]. $\mathcal{R}(\mathbf{v})$ then rewrites as the sum of local terms related to vertices \mathbf{v}_i . The purpose of the paper, however, is not to discuss their expression.

C. Optimization Algorithms

We now consider the optimization of criterion $\mathcal{J}(\mathbf{v})$, which is multimodal and nondifferentiable at some points. There are mainly two classes of methods for such optimization, as follows:

- 1) stochastic methods, e.g., simulated annealing for addressing nonconvex optimization;
- 2) deterministic iterative local methods [32].

Because criterion \mathcal{J} has several global minimizers that represent the same shape (see Remark 2) and to limit the computation burden, we select the latter methods and, particularly, first-order descent methods, based on the knowledge of the gradient $\nabla \mathcal{J}(\mathbf{v})$. Among them, we distinguish direct (on all components simultaneously) and block relaxation methods, where blocks are composed of the coordinates of each vertice \mathbf{v}_i (two scalar variables).

In the following, we summarize those methods and then emphasize their underlying difficulties for polygonal reconstruction. We denote by $\mathbf{v}^{(k)} = \{\mathbf{v}_1^{(k)}, \dots, \mathbf{v}_n^{(k)}\}$ the k th iteration estimate of the vertices, $\mathbf{p}^{(k)} = \mathcal{A}(\mathbf{v}^{(k)})$ its projections, and $\mathbf{g}^{(k)} = \nabla \mathcal{J}(\mathbf{v}^{(k)})$ for $k = 0, 1, \dots, K$. Calculation of those quantities will be discussed in Section III-D.

1) *Gradient Descent:* The unconstrained gradient descent method to optimize $\mathcal{J}(\mathbf{v})$ on \mathbb{R}^{2n} can be summarized by the following iterative scheme:

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} - s^{(k)} \mathbf{g}^{(k)} \quad (8)$$

where $s^{(k)}$ is the descent step at iteration k . For instance, $s^{(k)}$ can be selected as the minimizer of $\mathcal{J}_k : s \rightarrow \mathcal{J}(\mathbf{v}^{(k)}) - s \mathbf{g}^{(k)}$ on \mathbb{R}_+ and performed using a cubic approximation of \mathcal{J}_k . The approximate line minimization is thus done analytically and involves additional calculation of $\mathcal{J}_k(s)$ at two points s_1 and s_2 to evaluate the cubic function, since $\mathcal{J}_k(s=0) = \mathcal{J}(\mathbf{v}^{(k)})$ and $\partial \mathcal{J}_k / \partial s (s=0) = -\|\mathbf{g}^{(k)}\|^2$ are already known.

Except for the computation of $\mathbf{p}^{(k)}$ and $\mathbf{g}^{(k)}$, the difficulty is due to the optimization domain \mathbb{A}_n , which is most often nonconvex. Thus, classic constrained versions of the gradient descent are not applicable [32]. However, the heuristic algorithm that links the unconstrained version and a test of acceptability of estimates $\mathbf{v}^{(k+1)}$ turns out to be efficient. Testing (5) involves the search for the intersection of all couples of edges of $\mathbf{v}^{(k+1)}$.

2) *Block Relaxation:* Also known as Gauss Seidel (GS) in numerical analysis or iterated conditional descent in statistical tomography [4], [32], the block relaxation algorithm is well adapted since the update of the projections $\mathcal{A}(\mathbf{v})$, and, hence, of $\mathcal{J}(\mathbf{v})$ is very straightforward when only one vertice is displaced (see Section III-D). The algorithm, summarized on Table I, is based on the alternate optimizations of $\mathcal{J}(\mathbf{v}) = \mathcal{J}(\mathbf{v}_i; \mathbf{v}_j, j \neq i)$ w.r.t. $\mathbf{v}_i \in \mathbb{R}^2$. Practically, we use a hybrid version of the GS algorithm, replacing marginal optimizations of $\mathcal{J}(\mathbf{v})$ by the corresponding gradient-based iterations, for all i in $\{1, \dots, n\}$

$$\mathbf{v}_i^{(k+1)} = \mathbf{v}_i^{(k)} - s_i^{(k)} \mathbf{g}_i^{(k)} \quad (9)$$

TABLE I
BLOCK RELAXATION PROCEDURE: ITERATION k . FOR SIMPLICITY REASONS, WE SUCCESSIVELY VISIT $\mathbf{v}_1, \dots, \mathbf{v}_n$

```

/* Beginning of iteration k: we know  $\mathbf{v}^{(k)}$  and  $\mathbf{p}^{(k)} \equiv \mathcal{A}(\mathbf{v}^{(k)})$  */
Set  $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)}$ .
For  $i = 1, \dots, n$ ,
    Estimate  $\mathbf{v}_i^{(k+1)}$  from  $\mathbf{v}_1^{(k+1)}, \dots, \mathbf{v}_{i-1}^{(k+1)}, \mathbf{v}_i^{(k)}, \dots, \mathbf{v}_n^{(k)}$ .
    Update  $\mathbf{p}^{(k+1)}$  by computing the projection of two triangles,
        composed of  $\mathbf{v}_{i-1}^{(k+1)}, \mathbf{v}_{i+1}^{(k)}$ , and the former/new positions of  $\mathbf{v}_i$ , i.e.,  $\mathbf{v}_i^{(k)}$  and  $\mathbf{v}_i^{(k+1)}$ .
End For.
Compute  $\mathcal{J}(\mathbf{v}^{(k+1)})$  from  $\mathbf{p}^{(k+1)} = \mathcal{A}(\mathbf{v}^{(k+1)})$ .
/* Acceptability test for  $\mathbf{v}^{(k+1)} = \{\mathbf{v}_i^{(k+1)}, i = 1, \dots, n\}$  */
Check condition (5).
If  $\mathbf{v}^{(k+1)}$  is not acceptable,
    End of minimization, the final estimate is  $\hat{\mathbf{v}} = \mathbf{v}^{(k)}$ .
End If.
/* Termination test */
If  $\mathcal{J}(\mathbf{v}^{(k)}) - \mathcal{J}(\mathbf{v}^{(k+1)}) < \varepsilon$ ,
    End of minimization, the final estimate is  $\hat{\mathbf{v}} = \mathbf{v}^{(k+1)}$ .
End If.
    
```

where $\mathbf{g}_i^{(k)} = \nabla \mathcal{J}(\mathbf{v}_i^{(k)}; \mathbf{v}_j, j \neq i) \in \mathbb{R}^2$ is the gradient of \mathcal{J} w.r.t. \mathbf{v}_i and the visiting order of vertices \mathbf{v}_i is chosen randomly uniform. Optimization steps $s_i^{(k)}$ are selected similarly to the gradient descent case, as the minimizers on \mathbb{R}_+ of $s \rightarrow \mathcal{J}(\mathbf{v}_i^{(k)} - s\mathbf{g}_i^{(k)}; \mathbf{v}_j, j \neq i)$. Their computation is done by the same cubic approximation.

The projection update, which motivates the use of the block algorithm, is also the leading point to the calculation of $\nabla \mathcal{J}(\mathbf{v})$; see Section III-D.

3) *Chosen Algorithm*: Both algorithms are local and then lead to similar results. Block relaxation is known to be more efficient on flat valleys of \mathcal{J} , but is more expensive. Although the computation cost is not a drastic factor in the 2-D case, we favored *gradient descent* in practice. Another possibility would be to first use the gradient algorithm to get a rough estimate of $\hat{\mathbf{v}}$ and then the relaxation algorithm.

The choice of initialization $\mathbf{v}^{(0)}$ is crucial since the minimization scheme is local. In [26], we have shown that prior reconstruction of a parametric shape $\hat{\mathcal{C}}^{(0)}$ with a low number of parameters (super-ellipse, harmonic contour, etc.) is a good compromise between the quality and rapidity of reconstruction. We estimate the contour parameters in the least square sense and then approximate $\hat{\mathcal{C}}^{(0)}$ by a polygon to obtain $\mathbf{v}^{(0)}$. Most often, an arc length description $\hat{\mathcal{C}}^{(0)}(s)$ is available, thus uniform sampling of the arc length ($s = s_1, \dots, s_n$) provides $\mathbf{v}^{(0)} = \{\hat{\mathcal{C}}^{(0)}(s_1), \dots, \hat{\mathcal{C}}^{(0)}(s_n)\}$. For both optimization algorithms, the maximum number of iterations is fixed ($K = 50$ for $n \approx 100$) and we terminate the algorithm if $\mathcal{J}(\mathbf{v}^{(k)}) - \mathcal{J}(\mathbf{v}^{(k+1)}) < \varepsilon$ for some arbitrary threshold ε .

4) *Computation Difficulties*: The direct calculation of the projections $\mathcal{A}(\mathbf{v})$ and their derivatives w.r.t. \mathbf{v}_i are the least requirements to the use of first-order descent methods. Among existing works on polygonal reconstruction, and more generally parametric contour reconstruction, we distinguish the following:

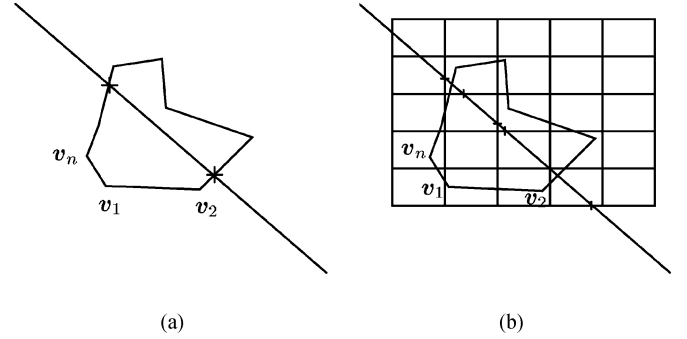


Fig. 4. Exact and approximate calculation of a closed-shape projection in the 2-D case. (a) Exact calculation based on the calculus of distances between intersections of the shape with the projection ray. (b) Approximate numerical calculation by pixel discretization of the 2-D image.

- 1) the ones that use a direct exact calculation of the projections and their partial derivatives (see [15] for the case of elliptic shapes);
- 2) the ones based on a pixel approximation of the binary image $f(x, y)$ derived from \mathcal{C}_v . Formation of the approximate image then involves, for each pixel element, the calculation of the pixel area enclosed in \mathcal{C}_v , and approximation of $\nabla \mathcal{A}(\mathbf{v})$ is done by an adjoint differentiation technique [17], [21]. Those techniques are also easily extendable to strip band integration.

Fig. 4 illustrates both techniques. Although for pixel approximation, the projection calculation is linear and straightforward, formation of the approximate image appears consuming in terms of computation time and memory storage, and especially in the prospect of 3-D extension. Because for polygons, a direct and simple calculation of projections is possible [27], we favor the first class of techniques, at the expense of technical developments regarding the calculation of the projection derivatives.

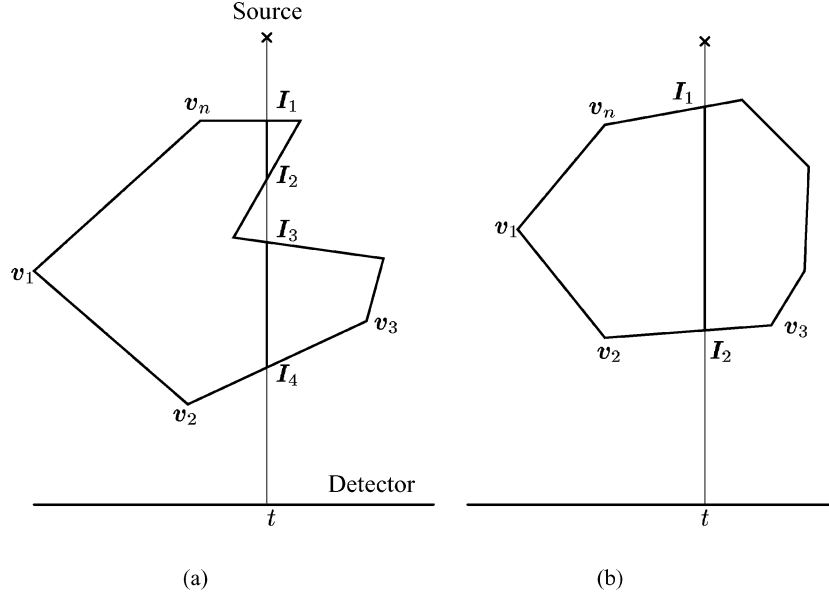


Fig. 5. (a) Evaluation of projection of a closed-contour C . In the regular case, the number of intersections between the projection ray and C is even. Positions I_1, I_3, \dots are the entries of the projection ray into C ; I_2, I_4, \dots are the exits. (b) For a convex shape, there are only two intersection points, and the projection value is equal to their distance.

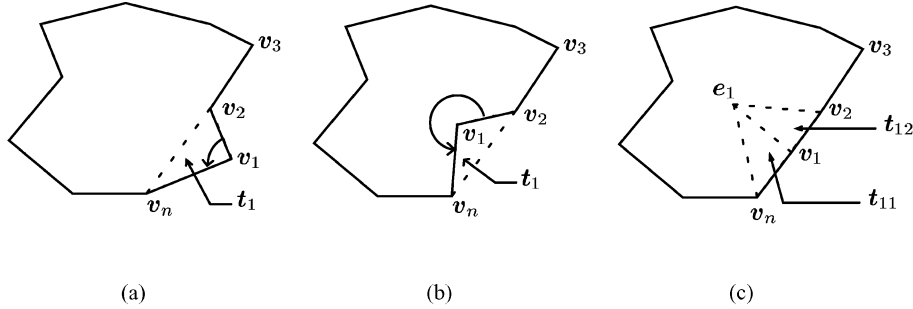


Fig. 6. Exterior status of vertex \mathbf{v}_1 of \mathbf{v} . (a) Exterior vertex: $\varepsilon_1 = 1$, the angle between $\mathbf{v}_1\mathbf{v}_2$, and $\mathbf{v}_1\mathbf{v}_n$ lays in $[0, \pi)$. (b) Interior vertex: $\varepsilon_1 = -1$, the angle between $\mathbf{v}_1\mathbf{v}_2$, and $\mathbf{v}_1\mathbf{v}_n$ lays in $[\pi, 2\pi)$. (c) Singular vertex: $\varepsilon_1 = 0$. The angle at \mathbf{v}_1 is flat and triangle \mathbf{t}_1 is empty.

D. Manipulation of Polygonal Projections

1) *Calculation of Projections*: For a fixed ray, the projection value of a polygon \mathbf{v} is given by (4). Clearly, $p_\theta(t)$ rewrites as the sum of the distances between the intersection points of the ray with contour $C_{\mathbf{v}}$ (see Fig. 5). If no vertex is located on the ray, the number of those points I_1, \dots, I_{2p} is even, and $p_\theta(t) = \sum_{j=1}^p \|I_{2j-1}I_{2j}\|$ [27]. The case of a convex polygon is even easier, yielding either 0 or 2 intersections.

Consequently, the projection calculation works as follows. We first register all intersection points I_k and then sort those points by their distance to a fixed position on the ray, e.g., the source position. If their number is even², we apply the above formula. If not, we do not search for the vertex(s) that lays on the projection ray, but process an approximate calculation using a parallel ray that is very close to the exact ray.

2) *Update of Projections*: We now quantify the update of $p_\theta(t)$ when a *single* vertex only, say \mathbf{v}_1 , is modified to any position. We assume that the new position $\mathbf{v}_1 + \delta\mathbf{v}_1$ still yields an acceptable polygon, denoted by $\mathbf{v} + \delta\mathbf{v} = \{\mathbf{v}_1 + \delta\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$.

²That does not necessarily mean that no vertex is located on the projection ray, although the special case of an even number of vertices laying on the projection ray is very scarce.

To consider the dependence of $p_\theta(t)$ upon \mathbf{v} , we rename the projection values $p_{\mathbf{v}}(t)$ and $p_{\mathbf{v}+\delta\mathbf{v}}(t)$, and define the exterior status of vertices.

Definition 1: Vertex \mathbf{v}_1 is said exterior (respectively interior) to \mathbf{v} if the angle between vectors $\mathbf{v}_1\mathbf{v}_2$ and $\mathbf{v}_1\mathbf{v}_n$ lays in $[0, \pi)$ (respectively, $[\pi, 2\pi)$).

Let ε_1 be the sign of this angle and \mathbf{t}_1 denote triangle $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_n\}$ (see Fig. 6). Since a displacement of \mathbf{v}_1 only modifies the edges containing \mathbf{v}_1 , i.e., $[\mathbf{v}_1, \mathbf{v}_2]$ and $[\mathbf{v}_n, \mathbf{v}_1]$ $p_{\mathbf{v}}(t)$ writes as the sum of $\varepsilon_1 p_{\mathbf{t}_1}(t)$ plus a function that does not depend on \mathbf{v}_1 . Therefore, the projection update simply deduces from the projection of \mathbf{t}_1

$$p_{\mathbf{v}+\delta\mathbf{v}}(t) - p_{\mathbf{v}}(t) = \tilde{\varepsilon}_1 p_{\tilde{\mathbf{t}}_1}(t) - \varepsilon_1 p_{\mathbf{t}_1}(t) \quad (10)$$

where $\tilde{\varepsilon}_1$ and $\tilde{\mathbf{t}}_1$ are the local characteristics of $\mathbf{v}_1 + \delta\mathbf{v}_1$ relative to the modified polygon $\mathbf{v} + \delta\mathbf{v}$. As a triangle projection is analytic and piecewise affine w.r.t. t (see Appendix I), update (10) appears simple and computationally efficient.

3) *Differentiation of Projections*: Differentiation of $p_{\mathbf{v}}(t)$ w.r.t. $\mathbf{v}_1 = [x_1, y_1]'$ straightforwardly derives from the previous paragraph. Indeed, computing $\partial p_{\mathbf{v}}(t)/\partial \mathbf{v}_1 =$

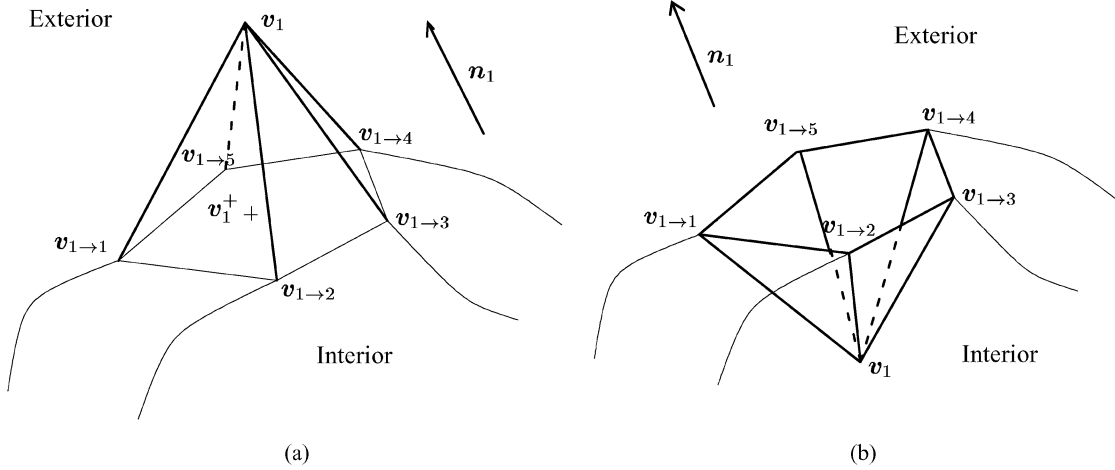


Fig. 7. Exterior status of vertex \mathbf{v}_1 . In both cases, initial formation of the faces guarantees that normal vectors $\mathbf{v}_1 \mathbf{v}_{1 \rightarrow i} \wedge \mathbf{v}_1 \mathbf{v}_{1 \rightarrow i+1}$ to the faces and \mathbf{n}_1 at vertex \mathbf{v}_1 are all oriented toward the exterior of the polyhedron. (a) Exterior vertex: $\varepsilon_1 = 1$. (b) Interior vertex: $\varepsilon_1 = -1$.

$[\partial p_{\mathbf{v}}(t)/\partial x_1, \partial p_{\mathbf{v}}(t)/\partial y_1]'$ is equivalent to updating $p_{\mathbf{v}}(t)$ following to an infinitesimal displacement of \mathbf{v}_1 . As the exterior property of \mathbf{v}_1 is invariant for infinitesimal displacements, (10) implies $\partial p_{\mathbf{v}}(t)/\partial \mathbf{v}_1 = \varepsilon_1 \partial p_{\mathbf{t}_1}(t)/\partial \mathbf{v}_1$. The special case in which \mathbf{v}_1 lays on edge $[\mathbf{v}_n, \mathbf{v}_2]$, that is $\varepsilon_1 = 0$, requires specific care and is illustrated in Fig. 6(c). We introduce a fictive vertex \mathbf{e}_1 that is interior to \mathbf{v} and form triangles $\mathbf{t}_{11} = \{\mathbf{v}_n, \mathbf{v}_1, \mathbf{e}_1\}$ and $\mathbf{t}_{12} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{e}_1\}$. The partial derivative then writes $\partial p_{\mathbf{v}}(t)/\partial \mathbf{v}_1 = \partial p_{\mathbf{t}_{11}}(t)/\partial \mathbf{v}_1 + \partial p_{\mathbf{t}_{12}}(t)/\partial \mathbf{v}_1$.

Finally, all cases reduce to the calculation of the derivatives of a triangle projection, which is done in Appendix I. Given a triangle $\mathbf{t} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ and a direction of projection θ , we provide analytic expressions of the projection $p_{\mathbf{t}}(t)$ and its derivative w.r.t. \mathbf{v}_1 . We show that $\partial p_{\mathbf{t}}(t)/\partial \mathbf{v}_1$ is defined for all t , except if \mathbf{v}_1 exactly projects on t . However, this case is practically scarce and allows definition of left and right derivatives. In conclusion, for a n vertex polygon \mathbf{v} , the derivative of $\mathcal{J}(\mathbf{v}) = \sum_{j,k} [d_j^k - p_{\theta_k}(t_j^k)]^2 + \lambda \mathcal{R}(\mathbf{v})$ w.r.t. \mathbf{v}_i is defined if no detector point t_j^k is equal to the geometric projection of \mathbf{v}_i , \mathbf{v}_{i-1} or \mathbf{v}_{i+1} onto the detector plane. $\partial \mathcal{J}(\mathbf{v})/\partial \mathbf{v}_i$ simply deduces from the projection derivatives, as the dependence of $\mathcal{J}(\mathbf{v})$ on projection values $p_{\theta_k}(t_j^k)$ is direct and quadratic, and functional $\mathcal{R}(\mathbf{v})$ is generally very simple.

IV. POLYHEDRAL RECONSTRUCTION

In this section, we study the binary polyhedron reconstruction problem. In image processing literature, deformable mesh models have been widely used for 3-D segmentation with the development of the finite element method [33]. However, their direct use for tomographic reconstruction purposes has been scarce. In fact, their reconstruction from projections is a difficult problem, since values of projections are not related to the vertice positions in a direct manner. For that application, recent contributions aim to directly estimate the position of vertices by statistical estimation procedures [21], [34], [35]. Remesh operations are then often needed [36], [37].

In the following, we only focus on the estimation of a polyhedron vertices, which raises several technical difficulties in itself. We extend the polygonal reconstruction method by introducing

the MAP estimator of the vertices. We then focus on the posterior energy optimization and its algorithmic counterparts, based on the manipulation of the polyhedron projections: their calculation, their update following to a vertice displacement, and their differentiation w.r.t. the vertices.

A. Acceptable Polyhedron Modeling

The primary difference between the 2-D and 3-D models is that a polyhedron is not fully specified by its vertices $\mathbf{v} = [\mathbf{v}'_1, \mathbf{v}'_2, \dots, \mathbf{v}'_n]'$, $\mathbf{v}_i \in \mathbb{R}^3$. Not only is there no natural ordering of vertices, but the shape description also requires a set of faces $\mathcal{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_q\}$. For simplicity reasons, we assume the following.

- 1) The number of vertices is fixed.
- 2) Faces are triangular.
- 3) the topology, i.e., the formation of faces is fixed by initial conditions; no remeshing is allowed.
- 4) The ordering of vertices is relative to the exterior of the polyhedron. If a face, say \mathcal{F}_i , is composed of vertices \mathbf{v}_{i_1} , \mathbf{v}_{i_2} , and \mathbf{v}_{i_3} , the normal vector to this face is defined by the cross product $\mathbf{v}_{i_1} \mathbf{v}_{i_2} \wedge \mathbf{v}_{i_1} \mathbf{v}_{i_3}$, and is oriented toward the exterior of the polyhedron.

Condition 4) allows, for any vertice \mathbf{v}_i , the definition of the ordered set $\mathbf{v}_{i \rightarrow 1}, \dots, \mathbf{v}_{i \rightarrow n_i}$ of its neighbors w.r.t. the exterior, where n_i is the number of neighbors (see Fig. 7). The faces that are adjacent to \mathbf{v}_i are then defined by triangles $\{\mathbf{v}_i, \mathbf{v}_{i \rightarrow j}, \mathbf{v}_{i \rightarrow j+1}\}$ for $j = 1, \dots, n_i$, where $\mathbf{v}_{i \rightarrow n_i+1}$ is set to $\mathbf{v}_{i \rightarrow 1}$. Conditions 3) and 4) imply that whatever the displacements of vertices, a face is always composed of the same vertices, and its normal vector *remains* oriented toward the exterior.

The shape parameterization finally reduces to the vertice positions $\mathbf{v}_1, \dots, \mathbf{v}_n$, and polyhedron reconstruction from projections is equivalent to the estimation of the vertices $\mathbf{v} \in \mathbb{R}^{3n}$. Similarly to the 2-D case, we define the set \mathbb{A}_n of unintersected, or acceptable closed polyhedra of faces \mathcal{F}

$$\mathbb{A}_n = \{\mathbf{v} \in \mathbb{R}^{3n}, \forall i, j \in \{1, \dots, q\}, i \not\sim j, \mathcal{F}_i \cap \mathcal{F}_j = \emptyset\} \quad (11)$$

where $i \not\sim j$ states that faces \mathcal{F}_i and \mathcal{F}_j are not adjacent.

In the literature of mesh segmentation, wider acceptability conditions have been derived, allowing a minimum distance between the faces [37]. In the field of surface reconstruction from projections, Battle *et al.* used a deformable polyhedron with a voxel approximation of the binary image [21], hence affording a heuristic check of acceptability conditions. For each voxel, they register the faces that intersect the voxel volume, and verify that any pair of intersecting faces ($\mathcal{F}_i, \mathcal{F}_j$) are neighbors. Although this test is practically efficient, it is not a sufficient condition to guarantee that $\mathbf{v} \in \mathbb{A}_n$, and, besides, the registration of intersections between all voxels and faces is a huge computation burden.

B. MAP Estimation

We estimate the vertice coordinates using the MAP estimator (7). With similar notation, the posterior energy $\mathcal{J}(\mathbf{v})$ is the sum of a fidelity to data term

$$\|\mathbf{d} - \mathcal{A}(\mathbf{v})\|^2 = \sum_{i,j,k} [d_{ij}^k - p_{\theta_k}(v_{ij}^k)]^2 \quad (12)$$

and a penalization term $\lambda\mathcal{R}(\mathbf{v})$, often selected to enhance the surface smoothness. Classic choices are based on the distance from vertices to the mass center of their neighbors, or their solid angle [21], [36], [38]. As in the 2-D case, criterion $\mathcal{J}(\mathbf{v})$ is generally multimodal and may have several global minimizers depending on the formation of the faces. We select local deterministic first-order descent schemes, as follows:

- 1) the gradient-based approach (8), optimizing \mathcal{J} w.r.t. all vertices simultaneously;
- 2) the block relaxation approach (9), in which blocks are composed of the vertice positions $\mathbf{v}_i \in \mathbb{R}^3$.

Initialization also affords the formation of the faces, which remain fixed during the iterations. Contrary to the 2-D case, the computation time is a drastic factor because of the large number of unknowns. Clearly, the relaxation scheme is greedier (see Section V for evaluation), and we then favor gradient descent. Nevertheless, a combination of both would be possible to successively explore tight then large valleys of criterion \mathcal{J} .

To carry out the optimization, we specify the direct calculation, update and differentiation of \mathcal{J} , which directly deduce from polyhedron projection calculation and differentiation algorithms. As those tools are extensions of polygon projection algorithms, we briefly summarize their outlines and focus on the special difficulties raised by the extension. For gradient descent, the acceptability test (11) can be done by checking that all couples of faces do not intersect. Instead, we choose to take advantage of the exact calculation of $\nabla\mathcal{J}(\mathbf{v}^{(k)})$ (valid for $\mathbf{v}^{(k)} \notin \mathbb{A}_n$) to propose a heuristic acceptability test. This test relies on the computation of the first-order finite differences $\partial\mathcal{J}(\mathbf{v}^{(k)})/\partial v_i$, and their comparison with the components of $\nabla\mathcal{J}(\mathbf{v}^{(k)})$. If the difference between both vectors is greater than an arbitrary threshold, set to $10^{-3}\|\partial\mathcal{J}(\mathbf{v}^{(k)})/\partial v_i\|$, we detect a self intersection of the shape. Although this test is not a sufficient condition to guarantee that $\mathbf{v}^{(k)} \in \mathbb{A}_n$, it yields very accurate results in practice.

C. Calculation and Update of Projections

Let us consider the projection of parameters (θ, t) of a polyhedron \mathbf{v} and rename $p_{\mathbf{v}}(t)$ the projection value. Similarly to the 2-D case, the direct calculation of $p_{\mathbf{v}}(t)$ is based on the search of intersection points $\mathbf{I}_1, \mathbf{I}_2, \dots$ between the projection ray and the polyhedron faces \mathcal{F}_j [27]. Since faces are triangular, this task is straightforward; for a face $\mathcal{F}_j = \{\mathbf{v}_{j_1}, \mathbf{v}_{j_2}, \mathbf{v}_{j_3}\}$, one simply needs to intersect the projection ray and the plane supporting \mathcal{F}_j and then check if the intersection point lays inside the triangle $\{\mathbf{v}_{j_1}, \mathbf{v}_{j_2}, \mathbf{v}_{j_3}\}$.

For regular configurations, in which the ray does not intersect one of the edges of \mathbf{v} , the number of intersection points $\mathbf{I}_1, \dots, \mathbf{I}_{2p}$ is even. Up to the sort of these points by their distance to a fixed location on the ray, e.g., the source position, the projection value is given by $p_{\mathbf{v}}(t) = \sum_{j=1}^p \|\mathbf{I}_{2j-1}\mathbf{I}_{2j}\|$. As in the 2-D case, singular configurations are handled using a parallel ray that is very close to the exact ray.

The projection update, which leads to projection differentiation, considers $p_{\mathbf{v}}(t)$ as a function of a particular vertice, say \mathbf{v}_1 . We, hereafter, quantify $p_{\mathbf{v}+\delta\mathbf{v}}(t) - p_{\mathbf{v}}(t)$, where $\mathbf{v} + \delta\mathbf{v}$ is the modified polyhedron composed of $\mathbf{v}_1 + \delta\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$.

1) *Local Vertice Attributes:* The exterior status of vertice \mathbf{v}_1 relies on the exterior normal vector \mathbf{n}_1 to the surface at \mathbf{v}_1 . As a polyhedral surface is not twice differentiable w.r.t. its arc lengths, we cannot properly define \mathbf{n}_1 . A good approximation is based on the connected neighbors $\mathbf{v}_{1 \rightarrow 1}, \dots, \mathbf{v}_{1 \rightarrow n_1}$ of \mathbf{v}_1 , and their mass center \mathbf{v}_1^+ [36]

$$\mathbf{n}_1 = u_1 \sum_{i=1}^{n_1} \mathbf{v}_1^+ \mathbf{v}_{1 \rightarrow i} \wedge \mathbf{v}_1^+ \mathbf{v}_{1 \rightarrow i+1} \quad (13)$$

where \wedge stands for the cross product, and $u_1 \in \mathbb{R}_+$ is chosen such that $\|\mathbf{n}_1\| = 1$ (see Fig. 7). The polyhedron construction guarantees that the sequence of vertices $\mathbf{v}_{1 \rightarrow i}$ is ordered relative to the exterior (see Section IV-A), and that \mathbf{n}_1 is also oriented toward the exterior.

Definition 2: Vertice \mathbf{v}_1 is said exterior (respectively interior) to polyhedron \mathbf{v} if and only if $\langle \mathbf{v}_1^+ \mathbf{v}_1, \mathbf{n}_1 \rangle \geq 0$ (respectively, ≤ 0), where $\langle \cdot, \cdot \rangle$ denotes the Euclidean dot product.

In the following, the sign of $\langle \mathbf{v}_1^+ \mathbf{v}_1, \mathbf{n}_1 \rangle$ is denoted by ε_1 .

2) *Update of Projections:* As a displacement of \mathbf{v}_1 yields the modification of the only faces containing \mathbf{v}_1 , we extract a local polyhedron $\mathbf{v}_{\partial 1}$ containing \mathbf{v}_1 , its neighbors $\mathbf{v}_{1 \rightarrow i}$, and a fictive vertice \mathbf{e}_1 for enclosure. This task is illustrated in Fig. 8, and the selection of \mathbf{e}_1 is discussed in Appendix III. As in the 2-D case, the difference between $p_{\mathbf{v}}(t)$ and $\varepsilon_1 p_{\mathbf{v}_{\partial 1}}(t)$ does not depend on \mathbf{v}_1 , and the projection update writes

$$p_{\mathbf{v}+\delta\mathbf{v}}(t) - p_{\mathbf{v}}(t) = \tilde{\varepsilon}_1 p_{\tilde{\mathbf{v}}_{\partial 1}}(t) - \varepsilon_1 p_{\mathbf{v}_{\partial 1}}(t) \quad (14)$$

where notation \sim refers to polyhedron $\mathbf{v} + \delta\mathbf{v}$. We perform analytic calculation of the projection of a local polyhedron, say $\mathbf{v}_{\partial 1}$, using its partition into the set $\mathbf{t}_{1l} = \{\mathbf{v}_1, \mathbf{v}_{1 \rightarrow l}, \mathbf{v}_{1 \rightarrow l+1}, \mathbf{e}_1\}$ of the tetrahedra formed of \mathbf{e}_1 and the faces that are adjacent to \mathbf{v}_1 . $p_{\mathbf{v}_{\partial 1}}(t)$ is then equal to the sum of their projections: $p_{\mathbf{v}_{\partial 1}}(t) = \sum_{l=1}^{n_1} p_{\mathbf{t}_{1l}}(t)$. Update (14) is finally very simple, since it only involves tetrahedra, whose projections are analytic; we refer the reader to Appendix II for the detailed presentation of their properties and calculation.

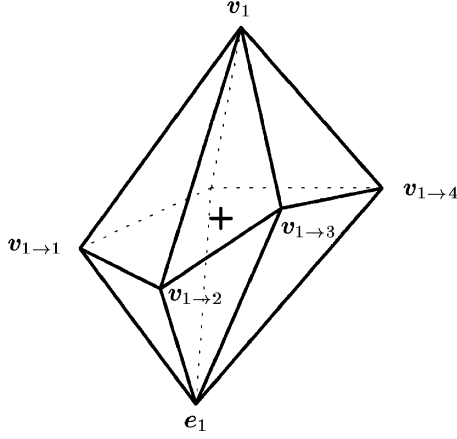


Fig. 8. Extraction of local polyhedron $\mathbf{v}_{\partial 1}$ containing \mathbf{v}_1 and its neighbors. Fictive vertex \mathbf{e}_1 is used as an enclosure of $\mathbf{v}_{\partial 1}$. Analytic calculation of the local polyhedron projections requires partition of $\mathbf{v}_{\partial 1}$ into the set of n_1 tetrahedra $\mathbf{t}_{1l} = \{\mathbf{v}_1, \mathbf{v}_{1 \rightarrow l}, \mathbf{v}_{1 \rightarrow l+1}, \mathbf{e}_1\}$.

D. Differentiation of Projections

Differentiation of projections is a direct consequence of their update. Indeed, an infinitesimal displacement of vertex \mathbf{v}_1 does not modify its exterior status ($\tilde{\varepsilon}_1 = \varepsilon_1$), and (14) yields

$$\frac{\partial \mathbf{p}_{\mathbf{v}}(t)}{\partial \mathbf{v}_1} = \varepsilon_1 \frac{\partial \mathbf{p}_{\mathbf{v}_{\partial 1}}(t)}{\partial \mathbf{v}_1} = \varepsilon_1 \sum_{l=1}^{n_1} \frac{\partial \mathbf{p}_{\mathbf{t}_{1l}}(t)}{\partial \mathbf{v}_1} \quad (15)$$

where partial derivatives are the vectors of \mathbb{R}^3 formed of the derivatives w.r.t. the coordinates x_1, y_1, z_1 of \mathbf{v}_1 . In consequence, the differentiation task reduces to the case where \mathbf{v} is a tetrahedron. In Appendix II, we provide the analytic expression of a tetrahedron projection, and then study its derivability w.r.t. its vertices. For a tetrahedron $\mathbf{t} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$, we show that $\partial \mathbf{p}_{\mathbf{t}}(t)/\partial \mathbf{v}_1$ is defined for all $t \in \mathbb{R}^2$, provided that the detector position t does not lay on a set of closed intervals $[t_i, t_j]$ delimited by the geometric projections t_i of vertices \mathbf{v}_i onto the detector plane (see Theorem 2).

In conclusion, the above conditions must be satisfied for all detector positions $t \equiv t_{ij}^k$ and all tetrahedra $\mathbf{t} = \mathbf{t}_{1l}$ to define the derivative of $\|\mathbf{d} - \mathcal{A}(\mathbf{v})\|^2$, and hence $\mathcal{J}(\mathbf{v})$ w.r.t. \mathbf{v}_1 .

V. SIMULATION RESULTS

We display 2-D and 3-D reconstructions obtained on a set of limited angle simulated data. Both experiments aim to reconstruct a synthetic shape \mathcal{C}^* from a limited number of noisy projections. The synthetic shape is a polygon of 40 vertices in the 2-D case and a polyhedron of 95 vertices and 186 triangular faces in the 3-D case. The simulated data \mathbf{d} are obtained according to (3) by first computing the exact direct projection $\mathcal{A}(\mathcal{C}^*)$ of the synthetic shape and then adding an i.i.d.-centered, white and Gaussian noise \mathbf{n} to the noiseless projections. The signal to noise ratio, defined by $\text{SNR} = 10 \log v_p/v_b$, where v_p and v_b are the empirical variances of the projection signal and the noise, is set to 20 dB.

A. Polygonal Reconstruction

This simulation illustrates the polygonal reconstruction method on a simple example. The data set represented in Fig. 9(a) is composed of $m = 4$ noisy projections of the synthetic shape

\mathcal{C}^* . The projection angles are uniformly distributed on $[0, \pi/2]$ and each projection is formed from 64 detector positions.

The reconstructed polygons involve $n = 20$ vertices and are obtained using the LS and MAP estimators. The regularization term corresponding to the MAP criterion is defined by

$$\mathcal{R}(\mathbf{v}) = \sum_{i=1}^n [1 + \cos a_i(\mathbf{v})]^2 \quad (16)$$

where $a_i(\mathbf{v}) \in (0, 2\pi)$ stands for the oriented angle between vectors $\mathbf{v}_i \mathbf{v}_{i-1}$ and $\mathbf{v}_i \mathbf{v}_{i+1}$. This choice affords penalization of sharp angles (high values of $\cos a_i(\mathbf{v})$), hence favoring local smoothness of \mathbf{v} . Vertex estimates are computed using the gradient descent scheme, and initialized by an elliptic shape, whose parameters are directly estimated from the data. Fig. 9(b) respectively displays estimates $\hat{\mathbf{v}}_{\text{ml}}$ and $\hat{\mathbf{v}}_{\text{map}}$, where hyperparameter λ is empirically chosen ($\lambda = 100$). The quality of the ML solution is acceptable for $n = 20$ but deteriorates as n grows larger. For high values of n , the polygonal shape lays closer to the initial ellipse, while presenting local irregularities [38]. MAP reconstructions appear accurate, and stable w.r.t. λ , as n is lower than 100. For $n = 20$, Fig. 9(c) shows that solution $\hat{\mathbf{v}}_{\text{map}}$ yields similar shapes for $\lambda \in [50, 600]$. Higher values of λ produce over-smoothed curved and do not allow the formation of nonconvex shapes.

B. Polyhedral Reconstruction

Polyhedral reconstruction is processed from a set of nine limited angle projections $\mathbf{d} = \{\mathbf{d}_1, \dots, \mathbf{d}_9\}$. The unknown surface \mathcal{C}^* and the projection geometry are represented in Fig. 10(a), whereas the projection images \mathbf{d}_k , formed of 64×64 pixels, are plotted in Fig. 10(b). The projection angles $\boldsymbol{\theta}_k = (\theta_k, \varphi_k)$, printed in Table II, are limited in the sense that all angles φ_k are lower than $\pi/4$. Therefore, the data suffer from a lack of information in the directions that are orthogonal to the vertical axis z . Moreover, surface \mathcal{C}^* is highly nonconvex, making its reconstruction by a deformable parametric surface very difficult.

Polyhedron vertices are estimated using the MAP estimator and the following penalization:

$$\mathcal{R}(\mathbf{v}) = \sum_{i=1}^n \left[1 + \cos \left(\frac{a_i(\mathbf{v})}{2} \right) \right]^2 \quad (17)$$

where $a_i(\mathbf{v}) \in (0, 4\pi)$ stands for the solid angle at vertex \mathbf{v}_i . This functional penalizes sharp vertices of \mathbf{v} , for which $a_i(\mathbf{v})$ is close to either 0 or 4π [38]. Fig. 10(d) displays the polyhedron reconstruction ($n = 95$ vertices, $q = 186$ triangular faces, and $\lambda = 2$), obtained with the gradient descent scheme and the spherical harmonic surface (c) as initialization. The 16 parameters of the harmonic surface are directly estimated from the data in the LS sense. Hyperparameter λ is empirically chosen, in order to avoid self intersected polyhedra while preserving the surface smoothness. Under- and over-regularized solutions ($\lambda = 1$ and $\lambda = 5$) are illustrated in Fig. 11. For $\lambda = 1$, the optimization algorithm yields an ‘‘almost’’ degenerate solution $\mathbf{v}^{(k)}$. More precisely, at iteration $k+1$, the gradient descent algorithm yields a vertex estimate $\mathbf{v}^{(k+1)}$, which does not fulfill the self intersection test. Consequently, the descent algorithm terminates, and the solution is set to the k th vertex estimate $\mathbf{v}^{(k)}$. Setting $\lambda = 5$ forbids the formation of highly nonconvex parts in the reconstructed shape.

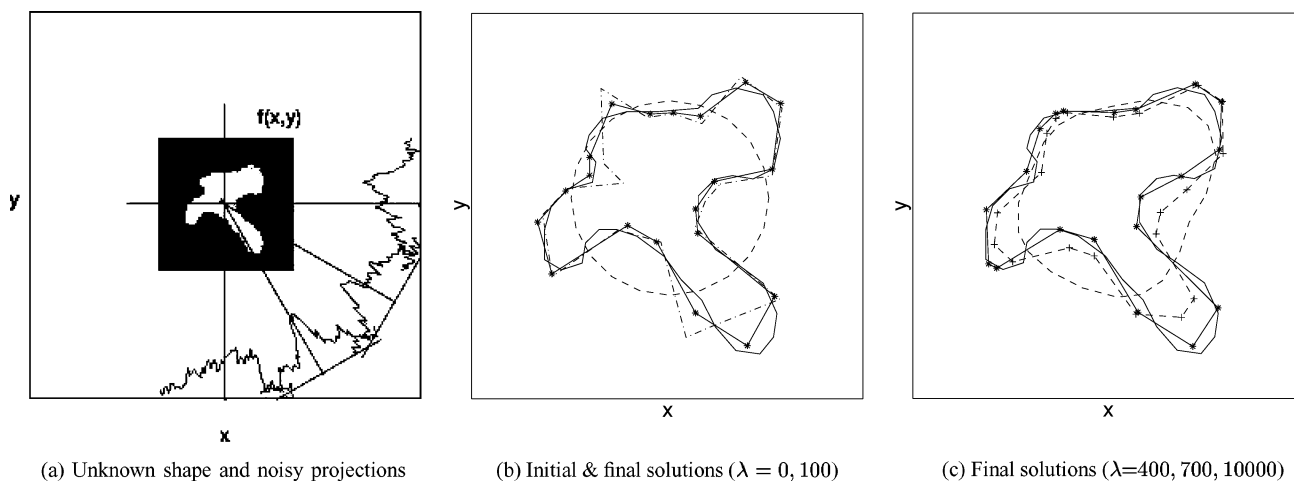


Fig. 9. Polygonal reconstructions. (a) Synthetic data are formed from four noisy projections of the unknown shape C^* (64 values per projection). The projection angles θ_k vary from 0 to $\pi/2$ and SNR = 20 dB. Reconstructed polygons are formed of $n = 20$ vertices and obtained using the gradient descent scheme. (b) Plot of elliptic initial solution $v^{(0)}$ (dashed curve) and final reconstructions obtained with $\lambda = 0$ (\hat{v}_{ml} , dashdot curve) and $\lambda = 100$ (\hat{v}_{map} , solid curve *). (c) Reconstructions corresponding to $\lambda = 400, 700, 10000$ (solid curve *, dashed curves +, and -, respectively). On plots (b) and (c), the unknown shape C^* is represented by the solid curve with no dots.

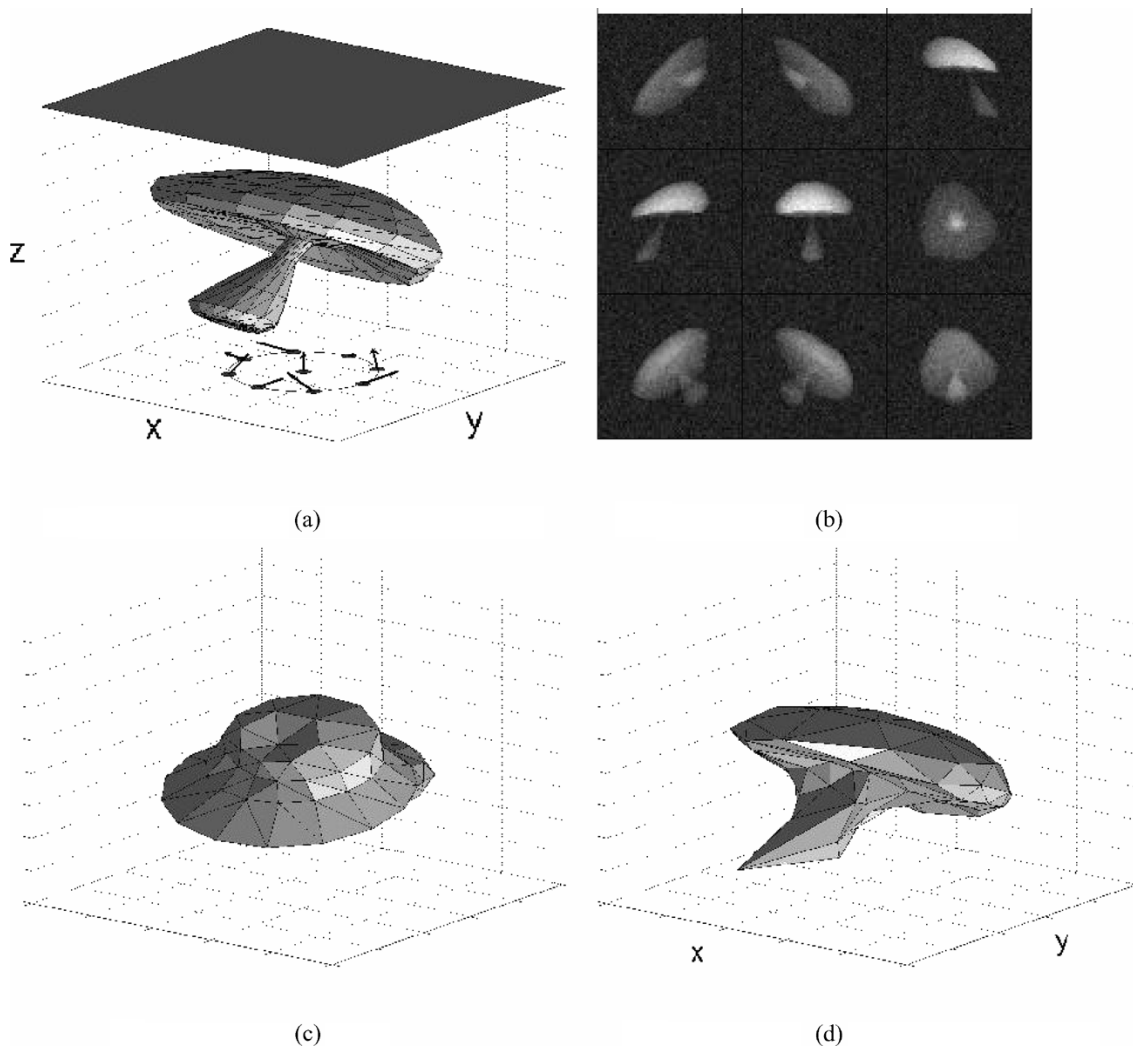


Fig. 10. Reconstruction results from nine simulated noisy projections. The projection images (b) are put together in a 3×3 "matrice," indexed by the projection angles (the first row corresponds to angles $\theta_1, \theta_2, \theta_3$, etc.). The harmonic model (c) depends on 16 parameters, estimated from d in the least-square sense. The reconstructed polyhedron (d) is obtained using the gradient descent scheme with (c) as an initialization. The number of iterations is $K = 10$. (a) Unknown surface C^* and projection geometry. (b) Synthetic data: nine projection images. (c) Harmonic reconstruction. (d) Polyhedral reconstruction ($n = 95, \lambda = 2$).

TABLE II
THREE-DIMENSIONAL PROJECTIONS: VALUES
OF ANGLES θ_k . ANGLE $\varphi = \arccos(\sqrt{3}/3)$

k	1	2	3	4	5	6	7	8	9
θ_k	$\pi/4$	$3\pi/4$	$-\pi/4$	$5\pi/4$	$-\pi/2$	$\pi/2$	0	π	0
φ_k	φ	φ	φ	φ	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$	0

Although both deterministic descent schemes, i.e., gradient and block relaxation yield similar results, the latter turns out to be very expensive. In Fig. 12, we plot for each scheme the time of reconstruction (CPU time), expressed in seconds, as a function of the number of vertices n for the data set of Fig. 10. In any case, and especially for large n , the overall CPU time is huge for block relaxation, thus favoring the gradient scheme and direct optimization on \mathbf{v} .

VI. CONCLUSION

We have studied general schemes for polygonal and polyhedral shape reconstruction from X-ray projection data. The proposed schemes directly estimate the position of the shape vertices \mathbf{v} using the MAP estimator. Minimization of posterior energy $\mathcal{J}(\mathbf{v})$ has been performed using deterministic first-order descent techniques. Among them, we have shown the interest of gradient techniques on the whole parameter vector \mathbf{v} and block relaxation techniques, where a block corresponds to a single vertice. Even though those algorithms are classic in numerical analysis, their use happens to be nontrivial for compact object reconstruction for two reasons: the nonconvexity of the minimization domain, and the direct computation of $\nabla \mathcal{J}$. The basis of our contribution is the direct computation, update, and differentiation of the projections with no use of a voxel approximation. These algorithms yield an efficient calculation method for $\nabla \mathcal{J}$, this quantity being required for the descent algorithms. In the 3-D case, the quality of reconstruction is similar for both techniques, but the computation burden is far higher for block relaxation. Therefore, we have selected gradient descent on \mathbf{v} . The self intersection test simply consists of the calculation of $\nabla \mathcal{J}$ and its comparison with its finite difference approximation.

Future work will primarily deal with the application of the polyhedron reconstruction method on real data in the field of nondestructive evaluation. On algorithmic parts, we think of using 2-D binary polygonal image sampling based on Arak fields [31] to perform stochastic optimization of $\mathcal{J}(\mathbf{v})$ in the polygonal case. Improvement of polyhedron reconstruction will aim to use nondifferentiable gradient techniques for the deterministic minimization of criterion \mathcal{J} [39]. Initial conditions can also be improved by the means of multiresolution techniques [34], [40]. They consist of successive reconstructions of polyhedra with an increasing number of vertices. First estimates are very quick and afford initialization of the latter optimization stages.

APPENDIX I

PROJECTION OF A TRIANGLE: CALCULATION AND DIFFERENTIATION

The aim of this section is to provide analytic expressions of the projection of angle θ of a triangle $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ and

its derivative w.r.t. \mathbf{v}_1 in the case it is defined. The projection is denoted by $p_{\mathbf{v}}(t; \theta)$ to express the dependence upon t and θ (Section I-A and I-E), or simply $p_{\mathbf{v}}(t)$ when $\theta = 0$.

A. Change of Basis

First, we show that all cases reduce to that of the vertical direction ($\theta = 0$) by the means of a change of basis.

We consider the projection of angle θ and denote by $\mathbf{u}_{\theta}^1 = [\cos \theta, \sin \theta]'$ the direction of detector and by $\mathbf{u}_{\theta}^2 = [-\sin \theta, \cos \theta]'$ the direction of projection. Let $\mathbf{R}(-\theta)$ be the rotation matrix of angle $-\theta$, which maps basis $\{\mathbf{u}_{\theta}^1, \mathbf{u}_{\theta}^2\}$ onto the standard basis $\{\mathbf{i}, \mathbf{j}\}$. The related change of basis writes $[X, Y]' = \mathbf{R}(-\theta)[x, y]'$, where

$$\mathbf{R}(-\theta) = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (18)$$

and (X, Y) represents the coordinates of a point (x, y) w.r.t. to basis $\{\mathbf{u}_{\theta}^1, \mathbf{u}_{\theta}^2\}$. Applying the change of basis to (4), the projection of angle θ rereads

$$p_{\mathbf{v}}(t; \theta) = \int_{\mathcal{L}_o(t)} f(x, y) dl = \int_{\mathcal{L}_o(t)} f(X, Y) dl = p_{\mathbf{w}}(t; 0) \quad (19)$$

where \mathbf{w} denotes triangle $\{\mathbf{R}(-\theta)\mathbf{v}_1, \mathbf{R}(-\theta)\mathbf{v}_2, \mathbf{R}(-\theta)\mathbf{v}_3\}$. In consequence, it is sufficient to study a triangle projection and its derivability in the case where $\theta = 0$. If $\partial p_{\mathbf{w}}(t; 0)/\partial \mathbf{w}_1$ is defined, then $\partial p_{\mathbf{v}}(t; \theta)/\partial \mathbf{v}_1$ also exists, and is equal to $\mathbf{R}(\theta)\partial p_{\mathbf{w}}(t; 0)/\partial \mathbf{w}_1$.

B. Vertical Projection of a Triangle

We now formulate the vertical projection $p_{\mathbf{v}}(t) = p_{\mathbf{v}}(t; 0)$ of triangle \mathbf{v} and distinguish two cases, whether \mathbf{v} contains a vertical edge (singular case) or not (regular case); see Fig. 13.

In the regular case, function $t \rightarrow p_{\mathbf{v}}(t)$ is continuous and piecewise affine. Up to a reordering of the vertices, we can assume that $\mathbf{v}_i = [x_i, y_i]'$ are such that $x_3 < x_1 < x_2$. The maximum P of $p_{\mathbf{v}}(t)$ is then reached at $t = x_1$, and

$$p_{\mathbf{v}}(t) = P \begin{cases} \frac{t-x_3}{x_1-x_3}, & \text{if } t \in (x_3, x_1] \\ \frac{x_2-t}{x_2-x_1}, & \text{if } t \in [x_1, x_2) \\ 0, & \text{if } t \notin (x_3, x_2) \end{cases} \quad (20)$$

where $P = p_{\mathbf{v}}(x_1)$. Denoting by ψ the angle $(\mathbf{i}, \mathbf{v}_3\mathbf{v}_2) \in (-\pi/2, \pi/2)$, we have $P = D/\cos \psi$, where D stands for the distance from \mathbf{v}_1 to line $(\mathbf{v}_2, \mathbf{v}_3)$.

In the singular case, one of the edges, say $[\mathbf{v}_1, \mathbf{v}_2]$ is vertical, and then $x_1 = x_2$. Function $p_{\mathbf{v}}(t)$ is piecewise affine, but not continuous at $t = x_1$. However, (20) still holds for all $t \neq x_1$.

We now study the differentiation of $p_{\mathbf{v}}(t)$ w.r.t. \mathbf{v}_1 in the regular case, and then generalize the result to the singular case.

C. Differentiation of Projections (Regular Case)

Differentiation of $p_{\mathbf{v}}(t)$ w.r.t. \mathbf{v}_1 gives rise to three cases, depending on the relative positions of x_1 , x_2 , and x_3 . The first case ($p_{\mathbf{v}}(x_1) \neq 0$) is plotted in Fig. 13(a), whereas the other two cases ($p_{\mathbf{v}}(x_1) = 0$) reduce to the study of $\partial p_{\mathbf{v}}(t)/\partial \mathbf{v}_2$ and $\partial p_{\mathbf{v}}(t)/\partial \mathbf{v}_3$ in the first case, up to a permutation of the vertices. Therefore, we assume that $x_3 < x_1 < x_2$ and then study the differentiation of $p_{\mathbf{v}}(t)$ w.r.t. each vertice separately. In particular,

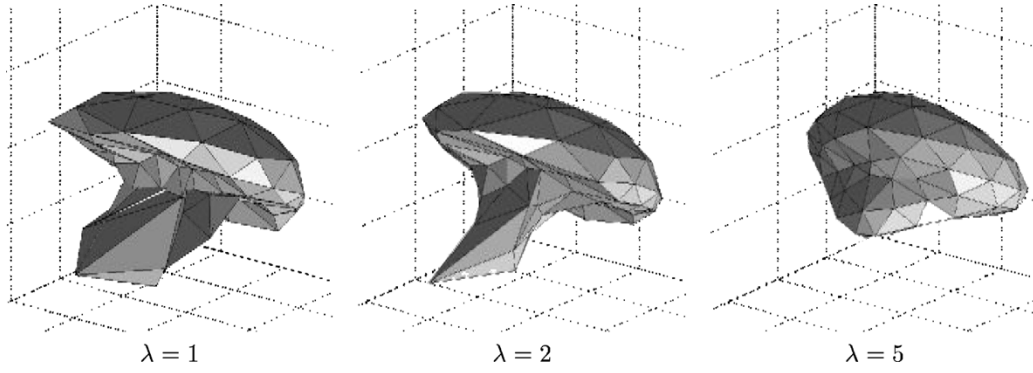


Fig. 11. Influence of λ on the reconstructed polyhedra. MAP reconstructions ($n = 95$ vertices) obtained with the gradient scheme, and the harmonic surface [Fig. 10(c)] as initialization.

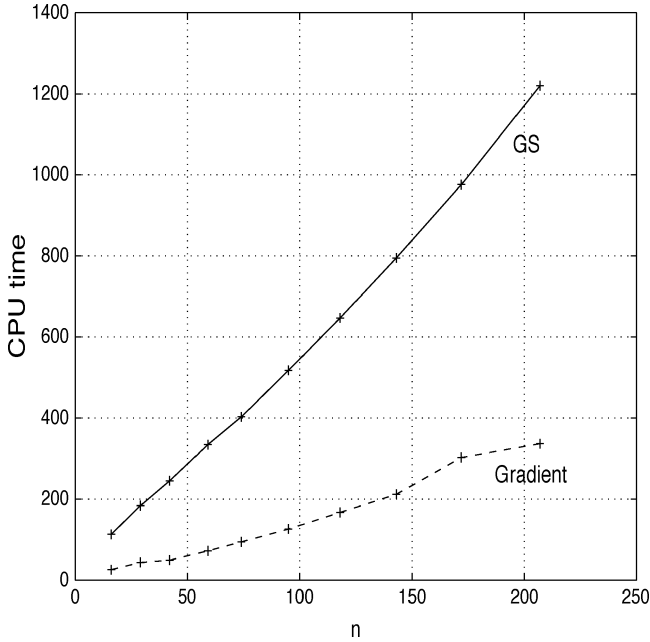


Fig. 12. Polyhedral reconstruction. Evaluation of the CPU time as a function of the number n of vertices for both local iterative algorithms: direct gradient descent on \mathbf{v} and block relaxation descent (GS). Reconstructions correspond to the data set plotted in Fig. 10(b), λ is empirically chosen, depending on n . Evaluation is expressed in seconds, and the number of iteration is $K = 10$.

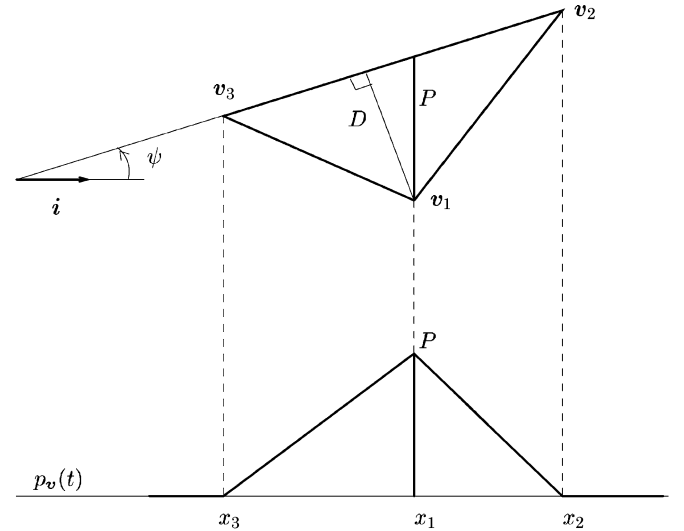
we put the stress on the differentiation of $P = p_{\mathbf{v}}(x_1)$ w.r.t. vertices \mathbf{v}_i , which directly leads to that of $p_{\mathbf{v}}(t)$ according to (20).

1) *Differentiation w.r.t. \mathbf{v}_1* : We can easily show that P is differentiable w.r.t. \mathbf{v}_1 , and that $\partial P / \partial x_1 = \tan \psi$ and $\partial P / \partial y_1 = 1$ if \mathbf{v}_1 lays above the line $(\mathbf{v}_2, \mathbf{v}_3)$, and -1 otherwise. Consequently, $p_{\mathbf{v}}(t)$ is differentiable for all $t \neq x_1$ and (20) yields

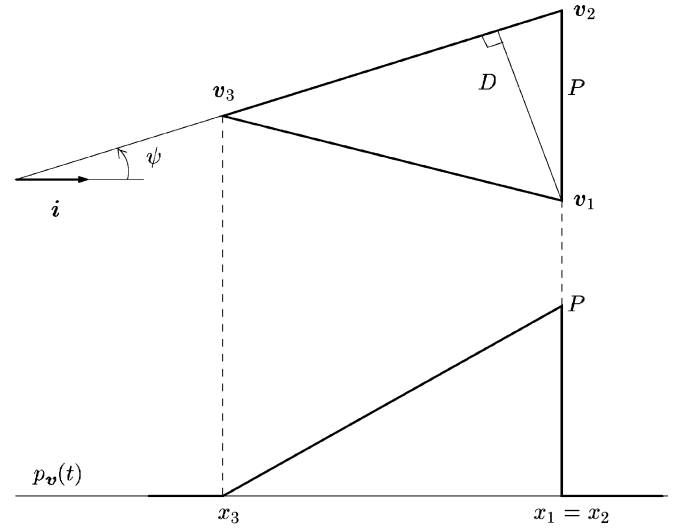
$$\frac{\partial p_{\mathbf{v}}(t)}{\partial \mathbf{v}_1} = \begin{cases} \frac{t-x_3}{x_1-x_3} \left[\frac{\partial P}{\partial \mathbf{v}_1} - \frac{P}{x_1-x_3} \mathbf{i} \right], & \text{if } t \in (x_3, x_1) \\ \frac{x_2-t}{x_2-x_1} \left[\frac{\partial P}{\partial \mathbf{v}_1} - \frac{P}{x_2-x_1} \mathbf{i} \right], & \text{if } t \in (x_1, x_2) \\ \mathbf{0}, & \text{if } t \notin (x_3, x_2). \end{cases} \quad (21)$$

This formulation allows the definition of left and right derivatives for $t = x_1$ by considering the limits of $\partial p_{\mathbf{v}}(t) / \partial \mathbf{v}_1$ as t tends to x_1^+ and x_1^- .

2) *Differentiation w.r.t. \mathbf{v}_2 and \mathbf{v}_3* : As $P = p_{\mathbf{v}}(x_1)$ is equal to $D / \cos \psi$, simple trigonometric calculation leads to $P =$



(a) Regular case: no edge is vertical.



(b) Singular case: x_1 is equal to x_2 or x_3 .

Fig. 13. Projection of a triangle $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ along vertical direction ($x_3 \leq x_1 \leq x_2$). Function $t \rightarrow p_{\mathbf{v}}(t)$ is continuous on \mathbb{R} in (a) the regular case and continuous on $\mathbb{R} \setminus \{x_1\}$ in (b) the singular case.

$|\det(\mathbf{v}_3 \mathbf{v}_1, \mathbf{v}_3 \mathbf{v}_2)| / (x_2 - x_3)$, where \det stands for the 2×2 matrix determinant operator. The latter determinant corresponds

to the half area of triangle \mathbf{v} . Hence, it is not equal to zero for nonflat triangles, and P is differentiable w.r.t. \mathbf{v}_2 and \mathbf{v}_3

$$\frac{\partial P}{\partial \mathbf{v}_2} = \varepsilon \frac{x_1 - x_3}{x_2 - x_3} \begin{bmatrix} \frac{y_3 - y_2}{x_2 - x_3} \\ 1 \end{bmatrix} \quad (22)$$

$$\frac{\partial P}{\partial \mathbf{v}_3} = \varepsilon \frac{x_1 - x_2}{x_3 - x_2} \begin{bmatrix} \frac{y_3 - y_2}{x_3 - x_2} \\ 1 \end{bmatrix} \quad (23)$$

where ε is equal to 1 if \mathbf{v}_1 lays below the line $(\mathbf{v}_2, \mathbf{v}_3)$ and -1 , otherwise. From these expressions, one can analytically compute $\partial p_{\mathbf{v}}(t)/\partial \mathbf{v}_2$ and $\partial p_{\mathbf{v}}(t)/\partial \mathbf{v}_3$ on the intervals $(x_3, x_1]$ and $[x_1, x_2)$, as is done in (21). In particular, $\partial p_{\mathbf{v}}(t)/\partial \mathbf{v}_i$ ($i = 2, 3$) is uniquely defined for all $t \neq x_i$. At $t = x_i$, left and right derivatives are, however, defined.

D. Differentiation of Projections (Singular Case)

Assuming that $x_3 \leq x_1 \leq x_2$, the singular case implies $x_1 = x_3$ or $x_1 = x_2$. Then, function $p_{\mathbf{v}}(t)$ is continuous on $\mathbb{R} \setminus \{x_1\}$ and the differentiation study done in Section I-C still holds for $t \neq x_1$. In particular, the computation of P and its derivatives w.r.t. \mathbf{v}_i , as well as those of $p_{\mathbf{v}}(t)$, are all valid for $t \neq x_1$ and x_i . If \mathbf{v}_i belongs to the vertical edge, x_1 and x_i are identical. In the alternative case, one can easily check that $P = p_{\mathbf{v}}(x_1)$ does not depend on \mathbf{v}_i . Hence, the extension of $\partial p_{\mathbf{v}}(t)/\partial \mathbf{v}_i$ at $t = x_1$ is possible by setting $\partial p_{\mathbf{v}}(x_1)/\partial \mathbf{v}_i = \mathbf{0}$. Finally, even in the singular case, differentiation of $p_{\mathbf{v}}(t)$ w.r.t. \mathbf{v}_i is possible for all $t \neq x_i$.

E. Conclusion

As a conclusion, the following theorem states the necessary and sufficient conditions for the differentiation of a triangle projection. The expression of the derivatives can be straightforwardly deduced from (19)–(23).

Theorem 1: Consider a nonflat triangle $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ and its projection of angle θ . Let t_i be the geometric projections of vertices \mathbf{v}_i onto the detector line ($t_i = \langle \mathbf{v}_i, \mathbf{u}_\theta^\perp \rangle$).

- 1) In the regular case (no edge is parallel to the direction of projection \mathbf{u}_θ^\perp), function $p_{\mathbf{v}}(t; \theta)$ is piecewise affine, continuous on \mathbb{R} , and differentiable w.r.t. \mathbf{v}_1 on $\mathbb{R} \setminus \{t_1\}$.
- 2) In the singular case, let $t_0 \in \{t_1, t_2, t_3\}$ be the geometric projection of the edge that is parallel to \mathbf{u}_θ^\perp . Function $p_{\mathbf{v}}(t; \theta)$ is piecewise affine, and continuous on $\mathbb{R} \setminus \{t_0\}$. It is differentiable w.r.t. \mathbf{v}_1 on $\mathbb{R} \setminus \{t_0, t_1\}$. If $t_0 \neq t_1$, the partial derivative is extendable by continuity at $t = t_0$.

At location $t = t_1$, “left” and “right” derivatives of $p_{\mathbf{v}}(t; \theta)$ w.r.t. \mathbf{v}_1 can, however, be defined by computing the limit values of function $t \rightarrow \partial p_{\mathbf{v}}(t; \theta)/\partial \mathbf{v}_1$ when t tends to t_1^+ and t_1^- .

APPENDIX II

PROJECTION OF A TETRAHEDRON: CALCULATION AND DIFFERENTIATION

We consider a tetrahedron $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ and its projection $p_{\mathbf{v}}(t)$ of angle $\theta = (\theta, \varphi)$. We study the computation of $p_{\mathbf{v}}(t)$ and its derivability w.r.t. a single vertice, say \mathbf{v}_1 . As for the 2-D projection of a triangle, function $t \rightarrow p_{\mathbf{v}}(t)$ is piecewise affine and continuous on \mathbb{R}^2 , provided that no face of \mathbf{v}

is parallel to the direction of projection.³ The support of $p_{\mathbf{v}}(t)$ is then equal to the convex hull of the geometric projections t_1, t_2, t_3 , and t_4 of vertices \mathbf{v}_i onto the detector plane (see Fig. 14). The support is then either triangular or quadrangular, depending on the relative positions of t_i . We now handle the calculation and differentiation of $p_{\mathbf{v}}(t)$ in both cases. We provide an analytic expression of $p_{\mathbf{v}}(t)$ and the necessary results to compute its derivatives.

A. Calculation of Projections for Triangular Support

If a location, say t_1 , lays inside the triangle $\{t_2, t_3, t_4\}$ formed of the other locations, the projection support is equal to the interior of this triangle; see Fig. 14(a). Function $p_{\mathbf{v}}(t)$ is then equal to zero for $t = t_2, t_3, t_4$ and is maximal at $t = t_1$. Denoting by P the maximum value, the projection is piecewise defined in the three inner triangles included in $\{t_2, t_3, t_4\}$

$$p_{\mathbf{v}}(t) = P \begin{cases} 1 - \frac{\det(\mathbf{t}_i, \mathbf{t}_j, \mathbf{t}_k)}{\det(\mathbf{t}_i, \mathbf{t}_j, \mathbf{t}_k)}, & \text{if } t \in \{t_1, t_i, t_j\} \\ 0, & \text{if } t \notin \{t_2, t_3, t_4\} \end{cases} \quad (24)$$

where \det denotes the 2×2 matrix determinant operator, and the bold notation refers to vectors of \mathbb{R}^2 . The three inner triangles $\{t_1, t_i, t_j\}$ are defined for $2 \leq i < j \leq 4$. Projection value $P = p_{\mathbf{v}}(t_1)$ can be directly computed from the vertices \mathbf{v}_i using $P = |\langle \mathbf{v}_2 \mathbf{v}_1, \mathbf{n}_1 \rangle| / \cos \psi$, where $\psi \in [0, \pi/2)$ denotes the angle between the projection ray \mathbf{u}_θ and the normal vector \mathbf{n}_1 to the plane $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$ (see Fig. 15).

B. Calculation of Projections for Quadrangular Support

If the locations t_i form a quadrangle, function $p_{\mathbf{v}}(t)$ is equal to zero for all positions $t = t_i$ and is maximal at the quadrangle center $t_0 \in \mathbb{R}^2$. Without loss of generality, we assume that t_1, t_2, t_3 , and t_4 are oriented w.r.t. t_0 in the counterclockwise order. The projection is then piecewise defined in the four inner triangles $\{t_0, t_i, t_{i+1}\}$ ($i = 1, \dots, 4$), where by circular extension, we set $t_5 \equiv t_1$

$$p_{\mathbf{v}}(t) = P \begin{cases} 1 - \frac{\det(\mathbf{t}_i, \mathbf{t}_j, \mathbf{t}_{i+1})}{\det(\mathbf{t}_i, \mathbf{t}_0, \mathbf{t}_{i+1})}, & \text{if } t \in \{t_0, t_i, t_{i+1}\} \\ 0, & \text{if } t \notin \{t_1, t_2, t_3, t_4\}. \end{cases} \quad (25)$$

The quadrangle center t_0 can be expressed as a function of locations t_1, t_2, t_3 , and t_4

$$\mathbf{t}_1 \mathbf{t}_0 = \frac{\det(\mathbf{t}_1 \mathbf{t}_2, \mathbf{t}_2 \mathbf{t}_4)}{\det(\mathbf{t}_1 \mathbf{t}_3, \mathbf{t}_2 \mathbf{t}_4)} \mathbf{t}_1 \mathbf{t}_3 \quad (26)$$

and $P = p_{\mathbf{v}}(t_0)$ is computed using the tetrahedron volume. Indeed, the volume of \mathbf{v} is equal to

$$\begin{aligned} \iiint_{\mathbb{R}^3} f(x, y, z) dx dy dz &= \cos \varphi \iint_{\mathbb{R}^2} p_{\mathbf{v}}(t) dt \\ &= \frac{P \cos \varphi}{6} \det(\mathbf{t}_1 \mathbf{t}_3, \mathbf{t}_2 \mathbf{t}_4) \end{aligned} \quad (27)$$

where $f(x, y, z)$ is the binary density function related to \mathbf{v} . The volume also writes as the mixed product $|\langle \mathbf{v}_1 \mathbf{v}_2, \mathbf{v}_1 \mathbf{v}_3 \wedge \mathbf{v}_1 \mathbf{v}_4 \rangle|/6$, leading to

$$P = \frac{|\langle \mathbf{v}_1 \mathbf{v}_2, \mathbf{v}_1 \mathbf{v}_3 \wedge \mathbf{v}_1 \mathbf{v}_4 \rangle|}{\cos \varphi \det(\mathbf{t}_1 \mathbf{t}_3, \mathbf{t}_2 \mathbf{t}_4)}. \quad (28)$$

³For simplicity, the case of singular tetrahedra, which is very similar to that of singular triangles in the 2-D case, will be eluded in the following.

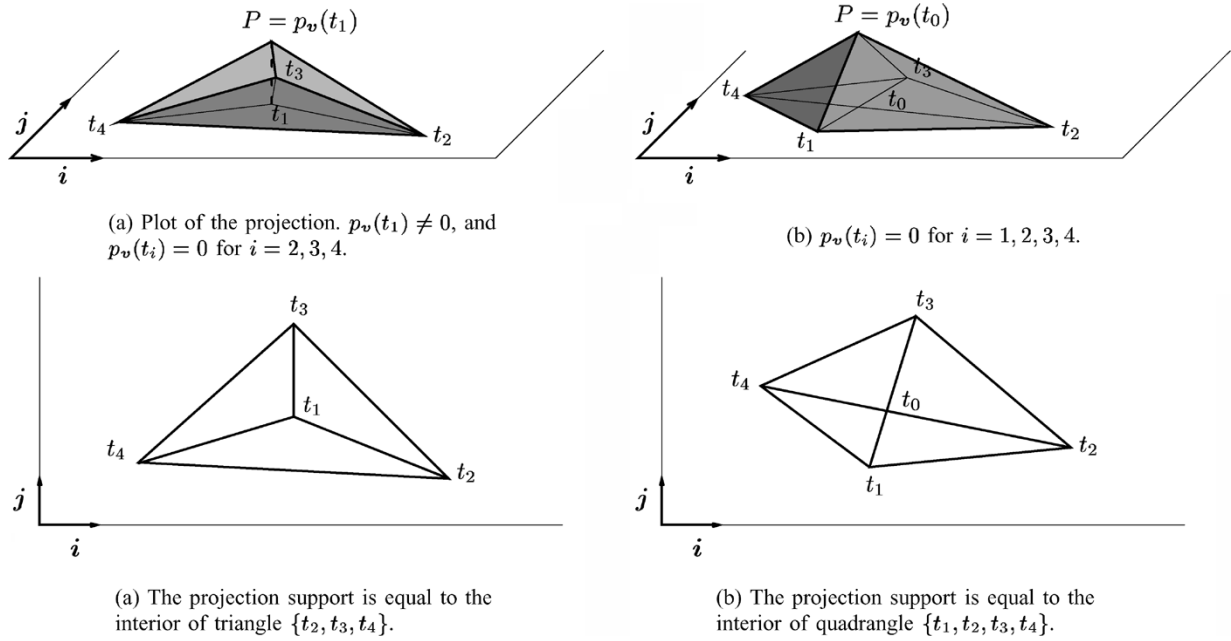


Fig. 14. Projection of a tetrahedron $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ onto the detector plane (upper figures). Function $t \rightarrow p_{\mathbf{v}}(t)$ is piecewise affine and continuous on \mathbb{R}^2 if no face of \mathbf{v} is parallel to the projection ray. The support of $p_{\mathbf{v}}(t)$ is equal to the convex hull of the geometric projections t_i of vertices \mathbf{v}_i onto the detector plane. Two cases appear, as the support is (a) triangular or (b) quadrangular. In the first case, one of the locations t_i , say t_1 , lays inside the triangle formed of the other three locations, and the maximum value P of $p_{\mathbf{v}}(t)$ is equal to $p_{\mathbf{v}}(t_1)$. In the second case, P is reached at the quadrangle center t_0 .

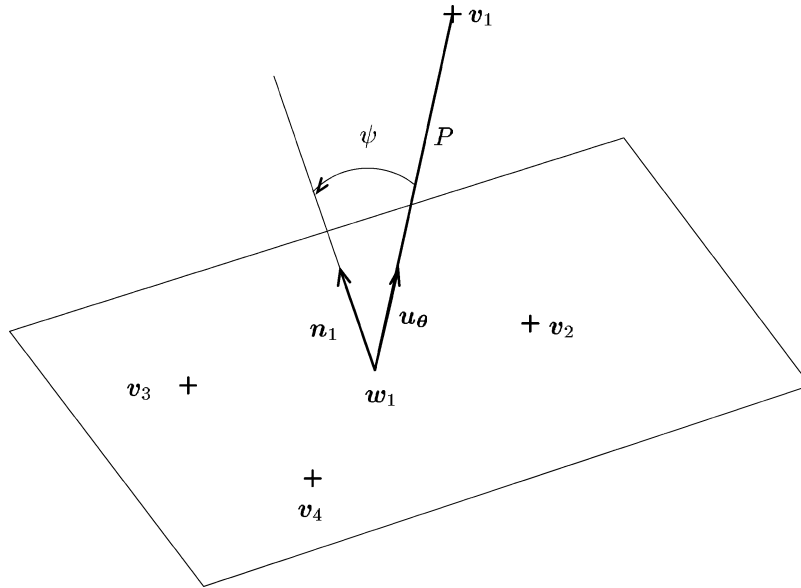


Fig. 15. Calculation of the projection of angle $\theta = (\theta, \varphi)$ of tetrahedron \mathbf{v} in case (a). The projection rays are directed by unit vector $\mathbf{u}_{\theta} = [\cos \theta \sin \varphi, \sin \theta \sin \varphi, \cos \varphi]^T$, and the projection value P is equal to the distance between \mathbf{v}_1 and its geometric projection \mathbf{w}_1 onto the plane $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$ along \mathbf{u}_{θ} . $P = |\langle \mathbf{v}_2 \mathbf{v}_1, \mathbf{v}_1 \rangle| / \cos \psi$, where \mathbf{n}_1 is the unit normal vector to the plane $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$, oriented toward the detector plane, and $\psi \in [0, \pi/2)$ stands for angle $(\mathbf{u}_{\theta}, \mathbf{n}_1)$.

C. Differentiation of Projections

Differentiation of $p_{\mathbf{t}}(t)$ w.r.t. \mathbf{v}_1 leads to three cases, as follows, whether

- 1) $p_{\mathbf{t}}(t)$ has a triangular support, and its maximal value is reached for $t \equiv t_1$;
- 2) $p_{\mathbf{t}}(t)$ has a triangular support, and its maximal value corresponds to another vertice, say $t \equiv t_2$;
- 3) $p_{\mathbf{t}}(t)$ has a quadrangular support.

Cases (1) and (3) correspond to Fig. 14(a) and (b) respectively, whereas case (2) reduces to the differentiation of $p_{\mathbf{t}}(t)$ w.r.t. \mathbf{v}_2 in the configuration of Fig. 14(a).

In all cases, differentiation of $p_{\mathbf{v}}(t)$ is done piecewise, on each inner triangular domain composing the projection support. Clearly, the partial derivatives of $p_{\mathbf{v}}(t)$ directly depend on those of t_i ($i \geq 1$) and P . The positions t_i are directly related to the corresponding vertices $\mathbf{v}_i = [x_i, y_i, z_i]^T$

$$t_i = \begin{bmatrix} x_i - (z_i - Z) \cos \theta \tan \varphi \\ y_i - (z_i - Z) \sin \theta \tan \varphi \end{bmatrix} \quad (29)$$

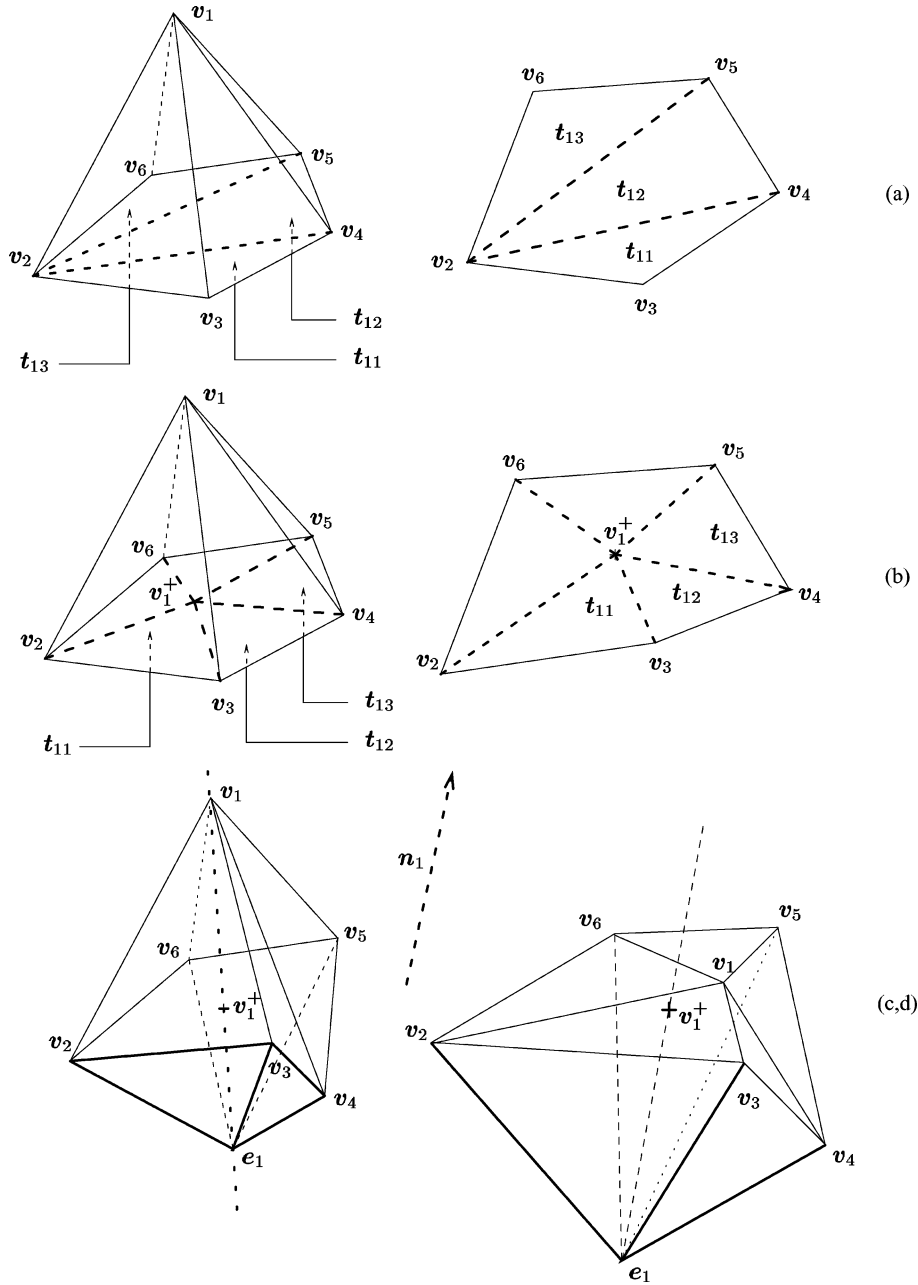


Fig. 16. Enclosure of $v_{\partial 1}$. Plots of local polyhedron $v_{\partial 1}$ and its orthogonal projection along normal vector \mathbf{n}_1 . For simplicity, neighbors of \mathbf{v}_1 are renamed $\mathbf{v}_2, \dots, \mathbf{v}_{n_1+1}$. (a) Case where the projections of the neighbors along \mathbf{n}_1 form a convex polygon. Its triangulation directly yields enclosure of $v_{\partial 1}$, and formation of $n_1 - 2$ tetrahedra t_{1l} . (b) Use of fictive vertex $\mathbf{e}_1 = \mathbf{v}_1^+$ to avoid flat tetrahedra, and formation of n_1 tetrahedra t_{1l} . (c,d) If the projected neighbors do not form a star-shaped polygon around \mathbf{v}_1^+ , \mathbf{e}_1 is selected on the line passing through \mathbf{v}_1^+ and directed by $\mathbf{v}_1 \mathbf{v}_1^+$ or $-\mathbf{e}_1 \mathbf{n}_1$, whether \mathbf{v}_1^+ and \mathbf{v}_1 are sufficiently distant or not.

where $z = Z$ refers to the detector plane. Consequently, the derivatives of the coordinates of t_i w.r.t. \mathbf{v}_i can be easily calculated. We now focus on the differentiation of P .

- 1) In cases (1) and (2), P is equal to $\varepsilon \langle \mathbf{v}_2 \mathbf{v}_1, \mathbf{n}_1 \rangle / \cos \psi$, where ε is set to 1 if \mathbf{v}_1 lays above the plane $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$, and -1 , otherwise (see Fig. 15). Calculation of $\partial P / \partial \mathbf{v}_1$ is direct, as both \mathbf{n}_1 and ψ do not depend on \mathbf{v}_1 and leads to $\partial P / \partial \mathbf{v}_1 = \varepsilon \mathbf{n}_1 / \cos \psi$. On the contrary, computing $\partial P / \partial \mathbf{v}_2$ requires the analytic expression of P as a function of \mathbf{v}

$$P = \varepsilon \frac{\langle \mathbf{v}_2 \mathbf{v}_1, \mathbf{v}_2 \mathbf{v}_3 \wedge \mathbf{v}_2 \mathbf{v}_4 \rangle}{\langle \mathbf{u}_\theta, \mathbf{v}_2 \mathbf{v}_3 \wedge \mathbf{v}_2 \mathbf{v}_4 \rangle}. \quad (30)$$

- 2) In case (3), $\partial P / \partial \mathbf{v}_1$ can be computed from (28).

We refer the reader to [38] for the complete calculation of the derivatives of P and those of $p_v(t)$ on each triangular domain composing the projection support. The conditions of derivability of $p_v(t)$ are now summarized for regular tetrahedra \mathbf{v} (no face of \mathbf{v} is parallel to the direction of projection).

Theorem 2: Consider the projection of angle θ of a regular tetrahedron $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$, and let $t_i \in \mathbb{R}^2$ be the geometric projections of vertices \mathbf{v}_i onto the detector plane. Then, the partial derivative of $p_v(t)$ w.r.t. \mathbf{v}_1 is defined if and only if t is not located on the following projected edges of \mathbf{v} :

- 1) $[t_1, t_i]$ for $i = 2, 3, 4$, in case (1);
- 2) $[t_1, t_i]$ and $[t_2, t_i]$ for $i = 2, 3, 4$, in case (2);

TABLE III
FORMATION OF FICTIVE VERTICE \mathbf{e}_1 FOR ENCLOSURE OF $\mathbf{v}_{\partial 1}$

Compute:

- mass centre \mathbf{v}_1^+ of the neighbors $\mathbf{v}_{1 \rightarrow j}$;
- normal vector \mathbf{n}_1 using (13);
- $\varepsilon_1 = \text{sign}(\langle \mathbf{v}_1^+ \mathbf{v}_1, \mathbf{n}_1 \rangle)$; /* Exterior status of \mathbf{v}_1 */
- dot products $p_j = \langle \mathbf{v}_1 \mathbf{v}_1^+, \mathbf{v}_1 \mathbf{v}_{1 \rightarrow j} \rangle$, $j = 1, \dots, n_1$.

If $(\forall j, p_j > 0)$,

/* Regular case, formation of a fictive vertice on $[\mathbf{v}_1; \mathbf{v}_1 \mathbf{v}_1^+]$ */

Do $\mathbf{e}_1 = \mathbf{v}_1 + \max(\{1\} \cup \{p_j / \|\mathbf{v}_1 \mathbf{v}_1^+\|^2\}) \mathbf{v}_1 \mathbf{v}_1^+$.

Else,

/* Singular case, generation of \mathbf{e}_1 on $[\mathbf{v}_1^+; -\varepsilon_1 \mathbf{n}_1]$ */

Do $\alpha = 2 \min\{\langle \mathbf{n}_1, \mathbf{v}_1^+ \mathbf{v}_1 \rangle, \langle \mathbf{n}_1, \mathbf{v}_1^+ \mathbf{v}_{1 \rightarrow 1} \rangle, \dots, \langle \mathbf{n}_1, \mathbf{v}_1^+ \mathbf{v}_{1 \rightarrow n_1} \rangle\}$. /* $\alpha < 0$ */

Do $\mathbf{e}_1 = \mathbf{v}_1^+ + \alpha \varepsilon_1 \mathbf{n}_1$.

End If.

3) both quadrangle diagonals and both projected edges that are adjacent to t_1 in case (3).

If t lays on one such projected edge, the directional derivative of $p_{\mathbf{v}}(t)$ w.r.t. the direction of projection \mathbf{u}_{θ} is, however, defined.

Remark 4: For a singular tetrahedron, let $[t_i, t_j]$ denote the geometric projection of the face that is parallel to \mathbf{u}_{θ} . Function $p_{\mathbf{v}}(t)$ is continuous on $\mathbb{R}^2 \setminus [t_i, t_j]$, and differentiable w.r.t. \mathbf{v}_1 at all locations t that do not lay on $[t_i, t_j]$ nor on the edges specified in Theorem 2.

APPENDIX III

GENERATION OF FICTIVE VERTICE FOR LOCAL POLYHEDRON ENCLOSURE

We hereafter define a heuristic scheme for the enclosure of a local polyhedron, say $\mathbf{v}_{\partial 1}$. As plotted in Fig. 16(a), enclosure does not necessarily need formation of a fictive vertice, but tetrahedra \mathbf{t}_{1l} are then flattened. Consequently, in any case, we decide to create a fictive vertice, denoted by \mathbf{e}_1 . Tetrahedra \mathbf{t}_{1l} are then formed of \mathbf{v}_1 , $\mathbf{v}_{1 \rightarrow l}$, $\mathbf{v}_{1 \rightarrow l+1}$, and \mathbf{e}_1 for $l = 1, \dots, n_1$. When the orthogonal projections of the neighbors $\mathbf{v}_{1 \rightarrow l}$ along normal vector \mathbf{n}_1 (on any plane) form a convex polygon, selecting $\mathbf{e}_1 \equiv \mathbf{v}_1^+$ as the mass center of vertices $\mathbf{v}_{1 \rightarrow l}$ is sufficient [see Fig. 16(b)]. For nonconvex cases, we cannot ensure that the polygon is star shaped around the projection of \mathbf{v}_1^+ . A good alternative is then to select \mathbf{e}_1 such that $\mathbf{e}_1 = \mathbf{v}_1 + \alpha \mathbf{v}_1 \mathbf{v}_1^+$ with $\alpha > 1$ if \mathbf{v}_1 and \mathbf{v}_1^+ are sufficiently distant, or else such that $\mathbf{e}_1 = \mathbf{v}_1^+ + \alpha \varepsilon_1 \mathbf{n}_1$, with $\alpha < 0$. The final selection scheme is detailed in Table III.

ACKNOWLEDGMENT

The authors would like to thank Drs. J. Idier and P. Ciuciu for their constant support and precious advice with this paper.

REFERENCES

- [1] K. D. Sauer, J. J. Sachs, and C. Klifa, "Bayesian estimation of 3-D objects from few radiographs," *IEEE Trans. Nucl. Sci.*, vol. 41, pp. 1780–1790, Oct. 1994.
- [2] S. Gautier, J. Idier, A. Mohammad-Djafari, and B. Lavayssière, "X-ray and ultrasound data fusion," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998, pp. 366–369.
- [3] J.-M. Dinten, "Tomographie à partir d'un nombre limité de projections: régularization par champs markoviens," Ph.D. dissertation, Univ. Paris-Sud, Orsay, France, Jan. 1990.
- [4] C. A. Bouman and K. D. Sauer, "A unified approach to statistical tomography using coordinate descent optimization," *IEEE Trans. Image Processing*, vol. 5, pp. 480–492, Mar. 1996.
- [5] P. Cinquin and B. Chalmoud, "Hip prosthesis design," *Lecture Notes Med. Inform.*, vol. 16, pp. 195–200, 1982.
- [6] Y. Bresler and A. Macovski, "Three-dimensional reconstruction from projections with incomplete and noisy data by object estimation," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-35, pp. 1139–1152, Aug. 1987.
- [7] J. A. Fessler and A. Macovski, "Object-based 3-D reconstruction of arterial trees from magnetic resonance angiograms," *IEEE Trans. Med. Imag.*, vol. 10, pp. 25–39, Mar. 1991.
- [8] P. Milanfar, W. C. Karl, and A. S. Willsky, "Reconstructing binary polygonal objects from projections: A statistical view," *Comput. Vis. Graph. Image Process.*, vol. 56, no. 5, pp. 371–391, Sept. 1994.
- [9] W. C. Karl, G. C. Verghese, and A. S. Willsky, "Reconstructing ellipsoids from projections," *CVGIP: Graph. Models Image Process.*, vol. 56, no. 2, pp. 124–139, 1994.
- [10] J. L. Prince and A. S. Willsky, "Convex set reconstruction using prior shape information," *Comput. Vis. Graph. Image Process.*, vol. 53, no. 5, pp. 413–427, Sept. 1991.
- [11] L. H. Staib and J. S. Duncan, "Parametrically deformable contour models," *Comput. Vis. Pattern Recognit.*, pp. 98–103, June 1989.
- [12] M. Kass, A. P. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.
- [13] E. Bardinet, L. D. Cohen, and N. Ayache, "Fitting 3-D data using superquadrics and free-form deformations," in *Proc. 12th Int. Conf. Pattern Recognition*, vol. 1, Jerusalem, Israel, Oct. 1994, pp. 79–83.
- [14] F. Santosa, "A level-set approach for inverse problems involving obstacles," *ESAIM: COCV*, vol. 1, pp. 17–33, Jan. 1996.
- [15] D. J. Rossi and A. S. Willsky, "Reconstruction from projections based on detection and estimation of objects—Parts I and II: performance analysis and robustness analysis," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-32, pp. 886–906, Aug. 1984.
- [16] Y. Amit and K. M. Manbeck, "Deformable template models for emission tomography," *IEEE Trans. Med. Imag.*, vol. 12, pp. 260–268, June 1993.
- [17] K. M. Hanson, G. S. Cunningham, G. R. J. Jennings, and D. R. Wolf, "Tomographic reconstruction based on flexible geometric models," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Austin, TX, Nov. 1994, pp. 145–147.
- [18] P.-C. Chiao, W. L. Rogers, N. H. Clinthorne, J. A. Fessler, and A. O. Hero, "Model-based estimation for dynamic cardiac studies using ECT," *IEEE Trans. Med. Imag.*, vol. 13, pp. 217–226, June 1994.
- [19] P.-C. Chiao, W. L. Rogers, J. A. Fessler, N. H. Clinthorne, and A. O. Hero, "Model-based estimation with boundary side information or boundary regularization," *IEEE Trans. Med. Imag.*, vol. 13, pp. 227–234, June 1994.
- [20] P. Milanfar, G. C. Verghese, W. C. Karl, and A. S. Willsky, "Reconstructing polygons from moments with connections to array processing," *IEEE Trans. Signal Processing*, vol. 43, pp. 432–443, Feb. 1995.

- [21] X. L. Battle, G. S. Cunningham, and K. M. Hanson, "Tomographic reconstruction using 3-D deformable models," *Phys. Med. Biol.*, vol. 43, pp. 983–990, 1998.
- [22] X. L. Battle, C. Le Rest, A. Turzo, and Y. J. Bizais, "3-D attenuation map reconstruction using geometrical models and free-form deformations," *IEEE Trans. Med. Imag.*, vol. 19, pp. 1–8, May 2000.
- [23] D. F. Yu and J. A. Fessler, "Edge-preserving tomographic reconstruction with nonlocal regularization," *IEEE Trans. Med. Imag.*, vol. 21, pp. 159–173, Feb. 2002.
- [24] H. Feng, W. C. Karl, and D. A. Castanon, "A curve evolution approach to object-based tomographic reconstruction," *IEEE Trans. Image Processing*, vol. 12, pp. 44–57, Jan. 2003.
- [25] A. Matheny and D. B. Goldof, "The use of three and four-dimensional surface harmonics for rigid and nonrigid shape recovery and representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 967–981, Oct. 1995.
- [26] C. Soussen and A. Mohammad-Djafari, "Closed surface reconstruction in X-ray tomography," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Thessaloniki, Greece, Oct. 2001, pp. 718–721.
- [27] J. Arvo, *Graphics Gems II*. New York: Academic, 1991.
- [28] K. M. Hanson, R. L. Bilisoly, and G. S. Cunningham, "Kinky tomographic reconstruction," *Proc. SPIE*, vol. 2710, pp. 156–166, 1996.
- [29] A. Mohammad-Djafari, "Binary polygonal shape image reconstruction from a small number of projections," *Elektrik*, vol. 5, no. 1, pp. 127–138, 1997.
- [30] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.
- [31] V. S. Borkar and S. K. Mitter, "Stochastic processes that generate polygonal and related random fields," *IEEE Trans. Inform. Theory*, vol. 42, pp. 606–617, Mar. 1996.
- [32] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.
- [33] A. J. Davies, *The Finite Element Method: A First Approach*. Oxford, U.K.: Clarendon, 1980.
- [34] C. Soussen and A. Mohammad-Djafari, "Multiresolution approach to the estimation of the shape of a 3-D compact object from its radiographic data," *Proc. SPIE*, pp. 150–160, July 1999.
- [35] J. Brankov, Y. Yang, and M. Wernick, "Tomographic image reconstruction using content-adaptive mesh modeling," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Thessaloniki, Greece, Oct. 2001, pp. 690–693.
- [36] A. J. Bulpitt and N. D. Efford, "An efficient 3-D deformable model with a self-optimising mesh," *Image Vis. Comput.*, vol. 14, pp. 573–580, 1996.
- [37] J.-O. Lachaud and A. Montanvert, "Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction," *Med. Image Anal.*, vol. 3, no. 2, pp. 187–207, 1998.
- [38] C. Soussen, "Reconstruction 3-D d'un objet compact en tomographie," Ph.d. dissertation, Univ. Paris-Sud, Orsay, France, Dec. 2000.
- [39] S. Alliney and S. A. Ruzinsky, "An algorithm for the minimization of mixed l_1 and l_2 norms with application to Bayesian estimation," *IEEE Trans. Signal Processing*, vol. 42, pp. 618–627, Mar. 1994.
- [40] G. S. Cunningham, I. Koyfman, and K. M. Hanson, "Improved convergence of gradient-based reconstructions using multi-scale models," *Proc. SPIE*, vol. 2710, pp. 145–155, 1996.



Charles Soussen was born in France in 1972. He received the degree from the École Nationale Supérieure en Informatique et Mathématiques Appliquées, Grenoble, France, and the Ph.D. degree in physics from the Laboratoire des Signaux et Systèmes, Université de Paris-Sud, Orsay, France, in 1996 and 2000, respectively.

He is currently an Assistant Professor with the Département de Physique, Université de Paris-Sud. His research interests are in inverse problems and image reconstruction.



Ali Mohammad-Djafari was born in Iran. He received the B.Sc. degree in electrical engineering from the Tehran Polytechnic, Tehran, Iran, in 1975, the M.Sc. degree from Supélec, Gif sur Yvette, France, in 1977, and the Ph.D. and the Doctorat d'État degrees in physics from the Université de Paris-Sud (UPS), Orsay, France, in 1981 and 1987, respectively.

Since 1984, he has held a permanent position at the Centre National de la Recherche Scientifique (CNRS), Gif-sur-Yvette, France, where he works in the Laboratoire des Signaux et Systèmes (L2S), Supélec. His main interests are in developing new probabilistic methods based on information theory, maximum entropy, and the Bayesian inference approaches for inverse problems, in general, and, more specifically, signal and image reconstruction and restoration. His application domains involve computed tomography, either for medical imaging or nondestructive evaluation.